Chapter 1

A Region-aided Colour Geometric Snake

1.1 Introduction

Deformable contour models or snakes are commonly used in image processing and computer vision due to their natural handling of shape variation and independence of operation (once initialised). A hypothesised contour, represented as a curve or surface, evolves under the influence of internal forces, external image dependent forces, and certain constraints, till it converges on the object(s) of interest.

Generally, there are two types of snakes, parametric snakes and geometric snakes. The parametric model minimises a deforming curve towards the pull of features such as edges and lines. The energy is composed of terms that control its smoothness and attract it to the object boundary. Although significant improvements have been made in this field over the last decade, parametric contours still suffer from imprecise shape representation. The geometric model of active contours, which avoids the need to parameterize the curve, has been hailed as the solution to topological problems. Geometric snakes are based on the theory of curve evolution and are numerically implemented via the level set algorithm. They are totally intrinsic, which means they can automatically handle topological changes without resorting to dedicated contour tracking and, unknown numbers of objects can be detected simultaneously. Furthermore, they can enjoy much larger capture areas than parametric snakes.

Whilst geometric or geodesic snakes go a long way in improving on parametric snakes, they still suffer from two significant shortcomings. First, they allow leakage into neighbouring image regions when confronted with weak edges; hereafter we refer to this as the weak-edge leakage problem. Second, they may rest at local maxima in noisy image regions. In this chapter, both of these problems are dealt with by introducing diffused region forces into the standard geometric snake formulation. The proposed method is referred to as the Region-

aided Geometric Snake or RAGS. It integrates gradient flows with a diffused region vector flow. The gradient flow forces supplant the snake with local object boundary information, while the region vector flow force gives the snake a global view of object boundaries. The diffused region vector flow is derived from the region segmentation map which in turn can be generated from any image segmentation technique. This chapter demonstrates that RAGS can indeed act as a refinement of the results of the initial region segmentation. It also illustrates RAGS' weak edge leakage improvements and tolerance to noise through various examples. Using colour edge gradients, RAGS will be shown to naturally extend to object detection in colour images. The Partial Differential Equations (PDEs) resulting from the proposed method will be implemented numerically using level set theory which enables topological changes to be dealt with automatically.

In Section 1.2, we review the geometric snake model, encompassing its strength and its shortcomings. Section 1.3 provides a brief overview of the geometric GGVF snake, also outlining its shortcomings. The former section is essential as RAGS' theory is built upon it, and the latter is necessary since we shall make performance comparisons to it. Section 1.4 presents the derivation of the RAGS snake including its level set representation. Then, in Section 1.5, the numerical solutions for obtaining the diffused region force and level set implementation of RAGS are introduced. Section 1.6 describes the extension of RAGS to vector-valued images, again showing the equivalent level set numerical representation. Since RAGS is independent of any particular region segmentation method, its description so far is not affected by the fact that no discussion of region segmentation has yet taken place! This happens next in Section 1.7 where the Mean Shift algorithm is employed as a typical, suitable method for obtaining a region segmentation map for use with RAGS. Following a brief summary of the RAGS algorithm in Section 1.8, examples and results illustrating the improvements obtained on noisy images and images with weak edges are presented in Section 1.9. This includes an application with quantitative results comparing the performance of RAGS against the standard geometric snake.

1.2 The Geometric Snake

Geometric active contours were introduced by Caselles et al. [1] and Malladi et al. [2] and are based on the theory of curve evolution. Using a reaction-diffusion model from mathematical physics, a planar contour is evolved with a velocity vector in the direction normal to the curve. The velocity contains two terms: a constant (hyperbolic) motion term that leads to the formation of shocks[1] from which more varied and precise representations of shapes can be derived, and a (parabolic) curvature term that smooths the front, showing up significant features and shortening the curve. The geodesic active contour, hereafter also referred to as the *standard geometric snake*, is now introduced. Let $C(x,t)$ be

[1] A discontinuity in orientation of the boundary of a shape; it can also be thought of as a zero-order continuity.

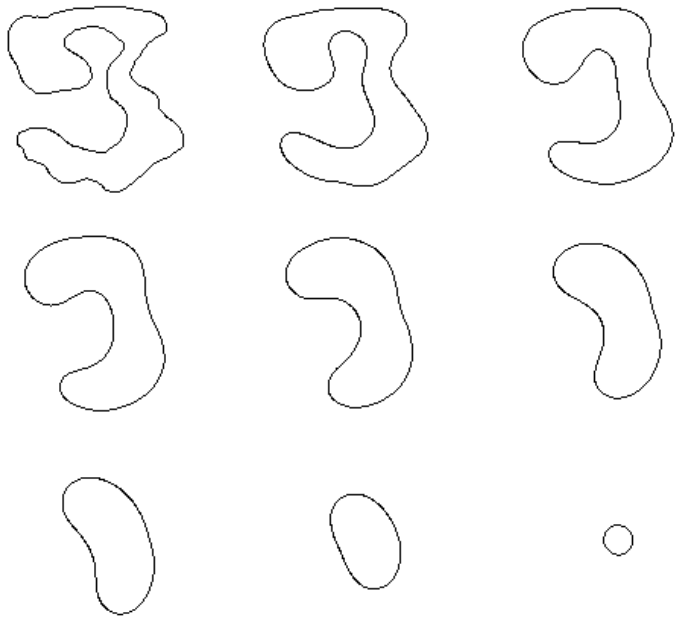

Figure 1.1: Motion under curvature flow: A simple closed curve will (become smoother and) disappear in a circular shape no matter how twisted it is.

a 2D active contour. The Euclidean curve shortening flow is given by

$$C_t = \kappa \vec{\mathcal{N}} \tag{1.1}$$

where t denotes the time, κ is the Euclidean curvature, and $\vec{\mathcal{N}}$ is the unit inward normal of the contour. This formulation has many useful properties. For example, it provides the fastest way to reduce the Euclidean curve length in the normal direction of the gradient of the curve. Another property is that it smooths the evolving curve (see Figure 1.1).

In [3, 4], the authors unified curve evolution approaches with classical energy minimization methods. The key insight was to multiply the Euclidean arc-length by a function tailored to the feature of interest in the image.

Let $I : [0, a] \times [0, b] \rightarrow \Re^+$ be an input image in which the task of extracting an object contour is considered. The Euclidean length of a curve C is given by

$$L := \oint |C'(q)| dq = \oint ds \tag{1.2}$$

where ds is the Euclidean arc-length. The standard Euclidean metric $ds^2 = dx^2 + dy^2$ of the underlying space over which the evolution takes place is modified

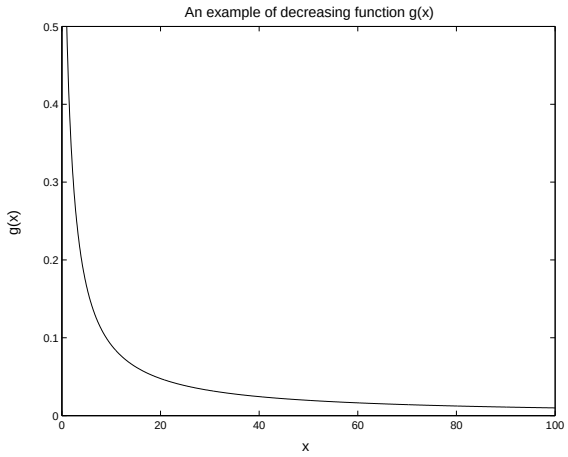

Figure 1.2: Plot of the monotonically decreasing function $g(x) = 1/(1+x)$.

to a conformal metric given by

$$ds_g^2 = g(|\nabla I(C(q))|)^2(dx^2 + dy^2) \tag{1.3}$$

where $g(.)$ represents a monotonically decreasing function such that $g(x) \to 0$ as $x \to \infty$, and $g(x) \to 1$ as $x \to 0$. A typical function for $g(x)$ can be:

$$g(x) = \frac{1}{1+x} \tag{1.4}$$

This is plotted in Figure 1.2. Using this metric, a new length definition in Riemannian space is given by

$$L_{\Re} := \int_0^1 g(|\nabla I(C(q))|)|C'(q)|dq. \tag{1.5}$$

Then the minimum path between two point in this metric is no longer necessary to be a straight line, which is the case in the standard Euclidean metric. The minimum path is now affected by the weighting function $g(.)$. Two distant points in the standard Euclidean metric can be considered to be very close to each other in this metric if there exists a route along which values of $g(.)$ are nearer to zero. The steady state of the active contour is achieved by searching for the minimum length curve in the modified Euclidean metric:

$$min \int_0^1 g(|\nabla I(C(q))|)|C'(q)|dq \tag{1.6}$$

Caselles et al. [4] have shown that this steady state is achieved by determining how each point in the active contour should move along the normal direction in

Figure 1.3: Motion under constant flow: it causes a smooth curve to evolve to a singular one.

order to decrease the length. The Euler-Lagrange of (1.6) gives the right-hand side of (1.7), i.e the desired steady state:

$$C_t = g(|\nabla I|)\kappa\vec{\mathcal{N}} - (\nabla g(|\nabla I|) \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} \tag{1.7}$$

Two forces are represented by (1.7). The first is the curvature term multiplied by the weighting function $g(.)$ and moves the curve towards object boundaries constrained by the curvature flow that ensure regularity during propagation. In application to shape modelling, the weighting factor could be an edge indication function that has larger values in homogeneous regions and very small values on the edges. Since (1.7) is slow, Caselles et al. [4] added a constant inflation term to speed up the convergence. The constant flow is given by $C_t = \vec{\mathcal{N}}$ showing each point on the contour moves in the direction of its normal and on its own can cause a smooth curve to evolve to a singular one (see Figure 1.3). However, integrating it into the geometric snake model lets the curvature flow (1.1) remain regular:

$$C_t = g(|\nabla I|)(\kappa + c)\vec{\mathcal{N}} - (\nabla g(|\nabla I|) \cdot \vec{\mathcal{N}})\vec{\mathcal{N}} \tag{1.8}$$

where c is a real constant making the contour shrink or expand to the object boundaries at a constant speed in the normal direction.

The second term of (1.7) or (1.8) depends on the gradient of the conformal factor and acts like a doublet, which attracts the active contour further to the feature of interest since the vectors of $-\nabla g$ point towards the valley of $g(.)$, the middle of the boundaries. This $-\nabla g$ increases the attraction of the active contour towards the boundaries. For an ideal edge, $g(.)$ tends to zero. Thus, it tries to force the curve to stop at the edge, but the convergence quality still

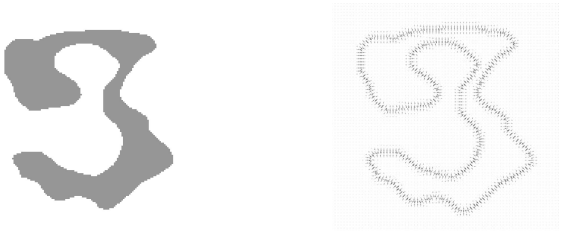

Figure 1.4: The doublet effect of the second term of equation (1.7)- The gradient vectors are all directed towards the middle of the boundary, which forces the snake into the valley of $g(.)$.

highly depends on this stopping term. If $g(.)$ is not small enough along edges, there will be an underlying constant force caused by c.

The geodesic or geometric active contour can be numerically implemented using Level Sets. This is demonstrated later in Section 1.4.4 when we deal with the extended formulation of the standard geometric snake into RAGS.

1.2.1 Examples of the standard Geometric Snake

The standard geometric snake has been applied successfully in many application areas, not least in the medical imaging arena. Figure 1.5-left shows an example of a geometric snake initialised in the stomach region of an abdominal section in a CT image. The final snake is shown in Figure 1.5-right. In the next example an extension of the geometric snake for colour images is shown in Figure 1.6, a thigh slice from the Visible Human Project. The figure on the left shows the initial snake as before, and the final converged snakes are shown on the right, demonstrating the topological adaptation of the snake's level set implementation. Note, the top snake has failed to fully converge. Hence, while adequate

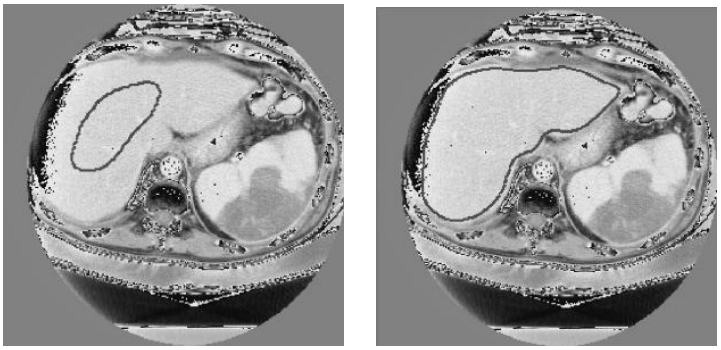

Figure 1.5: Example of geometric snake segmenting an inner boundary: recovery of the stomach region of an abdominal CT section - From [5], ©2003 IEEE.

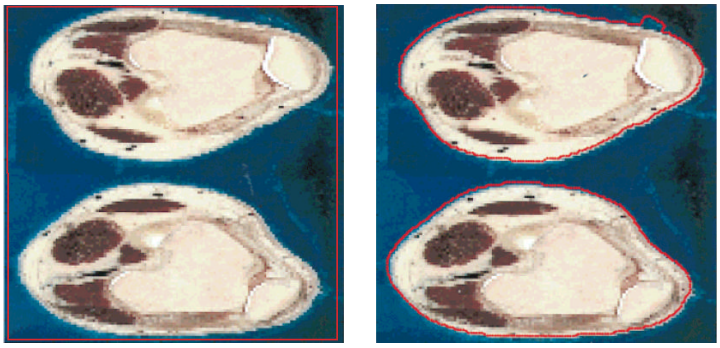

Figure 1.6: Example of a colour image - left: original image with initial snake - right: converged geometric snake (with a minor convergence problem in the top right corner of the snake) - From [6]. (Colour Slide)

for many situations, geometric snakes can suffer from certain shortcomings and the next section briefly deals with the nature of some such failings.

1.2.2 Shortcomings of the Geometric Snake

Geometric active contour models have the significant advantage over classical snakes that changes in topology due to the splitting and merging of multiple contours are handled in a natural way. However, they suffer in two specific ways:

1. They use only local information and hence are sensitive to local minima. This means they are attracted to noisy pixels and can fail to converge on the desired object when they rest at such strong 'features'. They fail to recognise, possibly weaker but true features further away in the image landscape, for lack of a better global understanding of the image. An example is shown in Figure 1.7-left.

2. The constant flow term makes the snake expand or shrink. It can speed up the convergence and push the snake into concavities easily when the objects have good contrast, i.e. when the gradient magnitudes at object boundaries are large. However, when the object boundary is indistinct or has gaps, the snake tends to leak through the boundary mainly because of this constant force. The second term in (1.8) is designed to attract the contour further close to the object boundary and also to pull back the contour if it leaks through the boundary, but the force may just not be strong enough since it still depends on the gradient values. It can not resolve the existence of a weak edge. Figure 1.7-right demonstrates this shortcoming of the standard geometric snake. The evolving of the snake is based on the gradient information, and as there is gradual change of the intensity, the contour leaks through.

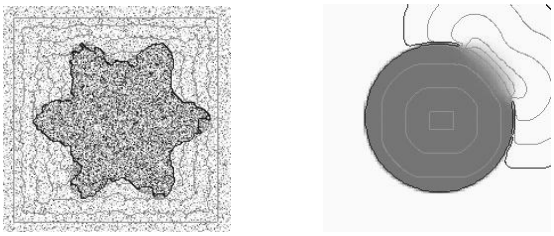

Figure 1.7: Noise sensitivity and weak-edge leakage problems. In each case the evolving snake is shown in a light colour and the final snake in a dark one.

The result of such failures is that the geometric snake will converge to a nonsensical form. Both these effects are demonstrated in Figure 1.8 where the cells contain fuzzy borders and strong but tiny dark 'granules' that have led the standard geometric snake astray (top-right image). The images in the bottom-row of Figure 1.8 show the region map used for the RAGS formulation outlined later in this chapter and the converged RAGS snakes. This figure also illustrates the power of the geometric snake in splitting to find multiple objects.

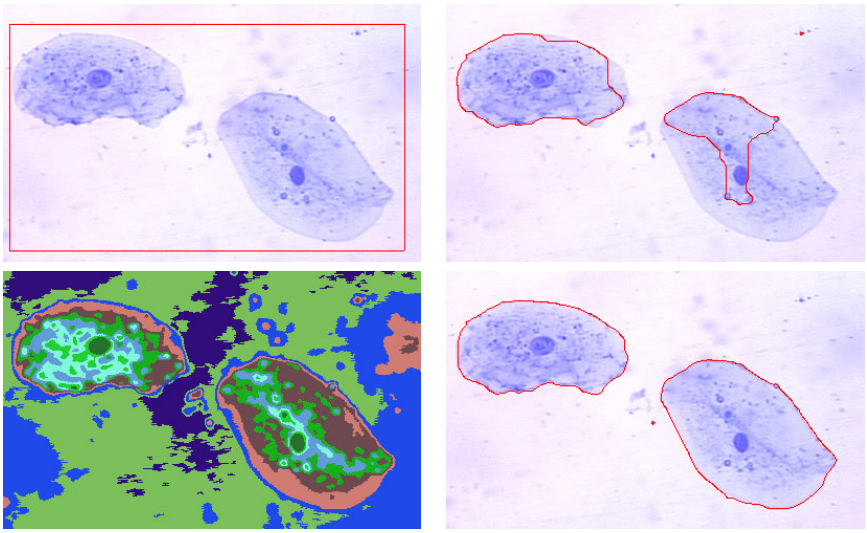

Figure 1.8: Multiple objects - top row: initial snake and standard geometric snakes - bottom row: region segmentation used by RAGS and converged RAGS snakes (*Original image courtesy of Dr. Douglas Kline, Department of Biological Sciences, Kent State University, US*). (Colour Slide)

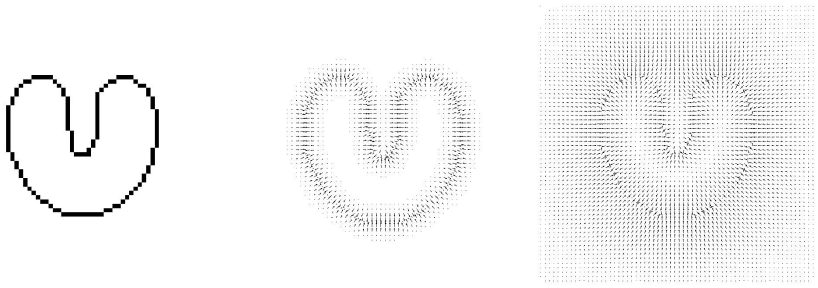

Figure 1.9: GVF Field compared to traditional potential force vector field - from left: A simple line-drawing U shape (binary) image, the traditional potential force vector field, and GVF Field. - From [8], ©2003 IEEE.

1.3 The Geometric GGVF snake

In this section we briefly introduce the geometric GGVF snake and consider its advantages and shortcomings. Later in the chapter, the GGVF snake will be used along with the standard geometric snake to make comparisons to the performance of RAGS.

The Gradient Vector Flow (GVF) active contour was first introduced by Xu and Prince [7] in a parametric framework. The authors proposed a new external force: *a diffusion of the gradient vectors of a greylevel or binary edge map* derived from the original image. The GVF goes some way towards forcing a snake into boundary concavities while providing a larger capture range due to its diffused gradient vector field. Figure 1.9-right shows the diffused gradient vectors for a simple object in 1.9-left. The traditional potential force is shown in 1.9-centre.

The same authors have also introduced the GGVF, a generalised GVF snake model. The GGVF improves the GVF by replacing the constant weighting factor with two spatially varying weighting functions resulting in a new external force field. The weighting factors provide a trade-off between the smoothness of the GVF field and its conformity to the gradient of the edge map. The result is that contours can converge into long, thin boundary indentations. The GGVF preserves clearer boundary information while performing vector diffusion, while the GVF will diffuse everywhere within image. As shown in Figure 1.10, the GGVF snake shows clear ability to reach concave regions.

Later in [10], Xu et al. showed the GGVF equivalence in a geometric framework. A simple bimodal region force generated as a two-class fuzzy membership function was added to briefly demonstrate weak-edge leakage handling. The geometric GGVF snake is useful when dealing with boundaries with small gaps. However, it's still not robust to weak edges, especially when a weak boundary is close to a strong edge, the snake readily steps through the weak edge and stops at the strong one. This is illustrated in Figure 1.11-left.

A further problem with the GGVF snake is that it does not always allow the

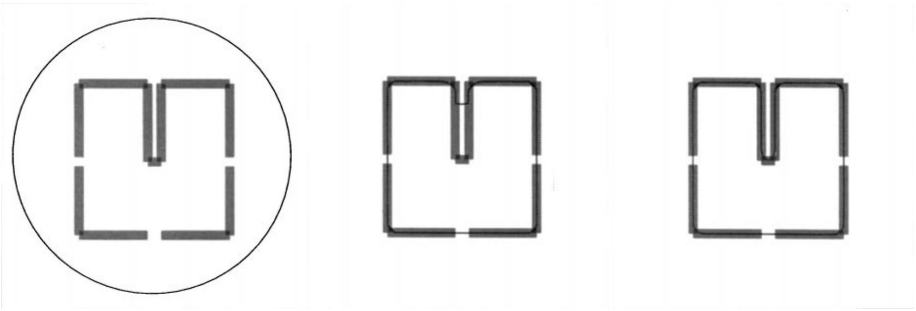

Figure 1.10: Concavity convergence comparison - from left: Initial snake, GVF snake result, and GGVF snake result - From [9].

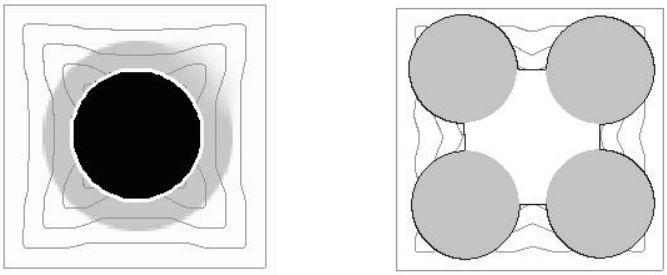

Figure 1.11: GGVF weaknesses - left: The GGVF snake steps through a weak edge towards a neighbouring strong one (final snake in white) - right: It also can encounter topological problems (final snake in black). The evolving snake is shown in a lighter colour in both cases.

detection of multiple objects. These topological problems arise, even though the GGVF snake was specified in the geometric model, when the vector field is tangent to the snake contour. In such cases there would be no force to push or pull it in the perpendicular direction (to the vectors). This effect is shown in Figure 1.11-right.

1.4 Region-aided Geometric Snake

We now begin to describe a novel approach to make the geometric snake much more tolerant towards weak edges and image noise. It comprises the integration of gradient flow forces with diffused region forces in the image resulting in the region-aided geometric snake, RAGS:

- The gradient flow forces supplant the snake with local object boundary information. They play a main role in all active contours[2].

[2]There are notable exceptions to this, e.g. [11].

- The region forces are based on the global image features and supplant the snake with global image information.

We show that this combination of forces not only improves the performance of the geometric snake towards weak edges, but also makes it more immune to noise. The PDE thus obtained, evolves an initial contour towards final convergence under the influence of both internal forces and boundary-regional image forces, and is implemented via level sets.

The proposed region force can be generated from any image segmentation technique. This means that while RAGS is independent of any particular segmentation technique, it is dependent on the quality of the regions produced. However, we show a good degree of tolerance to (reasonable) segmentation quality, and that our snake indeed acts as a refinement of the results of the initial region segmentation. Later in Section 1.7, we introduce the mean shift segmentation technique presented by Comaniciu and Meer in [12, 13] which is a very elegant method to generate region maps for this work. Results will be presented based on region maps obtained from both the under-segmentation and over-segmentation options of the software from Comaniciu and Meer.

1.4.1 Gradient flow force: a summary

As mentioned earlier, the gradient flows impose local constraints while the region force contributes global constraints. Within a homogeneous region of an image, measured by region segmentation, the snake evolves mainly according to gradient flows. The first gradient flow is the weighted length gradient flow, which is given by (1.7). It is composed of two terms. The first is the weighted curvature term, $g(|\nabla I|)\kappa\vec{\mathcal{N}}$, which smooths the active contour and also shrinks it. The second term, $(\nabla g(|\nabla I|)\cdot\vec{\mathcal{N}})\vec{\mathcal{N}}$, is on the normal factor of the gradient of the weighting function. Unlike the curvature, the vector field $\nabla g(|\nabla I|)$ is static. The direction and strength of this field depends on position only, independent of time and contour.

The second gradient flow, $g(|\nabla I|)c\vec{\mathcal{N}}$, is introduced by constant motion which locally minimises area (see [14] for proof). It helps the snake shrink or expand towards object boundaries and accelerates its convergence speed.

For all these forces, the weighting function g can be defined as any decreasing function of the image I edge map f such that $g \to 0$ as $f \to \infty$. When dealing with greylevel images, the solution (as used in this work) is straightforward:

$$f = |\nabla(Gauss * I)| \quad \text{and} \quad g = \frac{1}{1+f} \tag{1.9}$$

This monotonically decreasing nature is illustrated in Figure 1.2. As for colour images, the edge function f becomes a little more intricate (an example function will be presented in Section 1.6). However, the derivation of the decreasing function g can remain the same.

1.4.2 Diffused region force

The aim of generating a region force is to empower the snake with a global view of image features. A typical region segmentation method splits an image into several regions, giving the segmentation map S. From this, the region map R is generated by computing the gradient of S. The gradient computation is the same as the edge computation stage for generating gradient forces. Then, we compute the gradient ∇R of this region map, resulting in region constraints in the vicinity of the region boundaries. Having slithered across a homogeneous region reliant on the gradient flow forces, if the snake tries to step from one region into another, it must concur with the region force in ∇R since it breaks the region criteria, which probably indicates a leakage. The force field ∇R has vectors pointing towards the centre of the region boundaries. The capture area of this pure region force is quite small; only immediate areas close to region boundaries. The vectors need to be diffused further away from the region boundaries to create a larger capture field. To achieve this, we can diffuse ∇R resulting in region forces with a larger capture area along the region boundaries. Hence, the region force vector field $[\tilde{R}(z) = (u(z), v(z)), z = (x, y)]$ is obtained by solving the following equations:

$$\begin{cases} p(|\nabla R|)\nabla^2 u - q(|\nabla R|)(u - \nabla R_u) = 0 \\ p(|\nabla R|)\nabla^2 v - q(|\nabla R|)(v - \nabla R_v) = 0 \end{cases} \tag{1.10}$$

where ∇^2 is the Laplacian operator with dimensions u and v, $p(.)$ and $q(.)$ are weighting functions that control the amount of diffusion, and ∇R_u and ∇R_v are the components of vector field ∇R along the u and v directions[3]. The weighting functions are selected such that $p(.)$ gets smaller as $q(.)$ becomes larger with the desirable result that in the *proximity* of large gradients, there will be very little smoothing and the vector field will be nearly equal to the gradient of the region map. We use the following functions for diffusing the region gradient vectors:

$$\begin{cases} p(|\nabla R|) = e^{-(|\nabla R|/K)} \\ q(|\nabla R|) = 1 - p(|\nabla R|) \end{cases} \tag{1.11}$$

where K is a constant and acts as a trade-off between field smoothness and gradient conformity. The solution of (1.10) is the equilibrium state of the following partial differential equations:

$$\begin{cases} u_t = p(|\nabla R|)\nabla^2 u - q(|\nabla R|)(u - \nabla R_u) \\ v_t = p(|\nabla R|)\nabla^2 v - q(|\nabla R|)(v - \nabla R_v) \end{cases} \tag{1.12}$$

where u and v are treated as functions of time. These partial differential equations can be implemented using an explicit finite difference scheme. An iterative process can be set up, and guaranteed to converge with the following constraint

$$\Delta t \leq \frac{\Delta x \Delta y}{4 p_{max}} \tag{1.13}$$

[3]Theoretically, ∇R can be diffused in any two orthogonal directions, u and v, within the image domain. However, practically we will only choose x and y directions corresponding to image plane coordinates. Thus ∇R_u and ∇R_v are equal to $\frac{\delta R}{\delta x}$ and $\frac{\delta R}{\delta y}$ respectively.

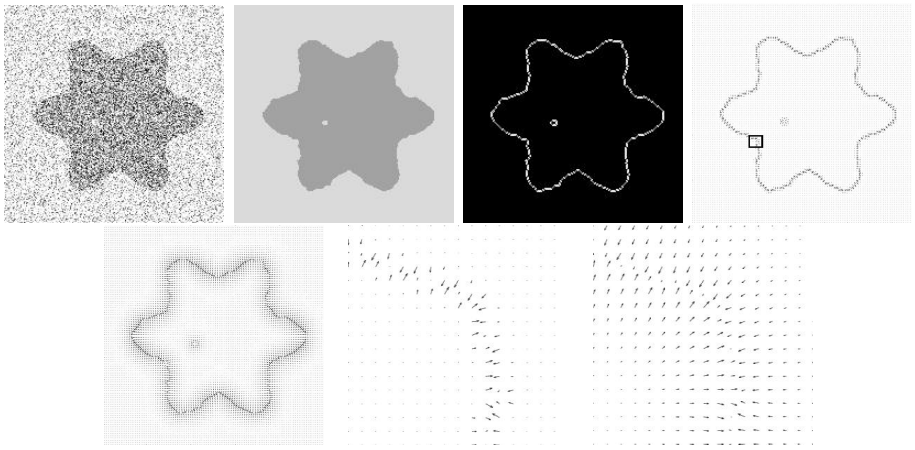

Figure 1.12: Region force diffusion - top row: A synthetic image with additive Gaussian noise, region segmentation map, region boundary map R, and gradient of the region map R (and a small selected area) - bottom row: diffused region vector field, and close-up views in the small selected area of the vectors in the gradient of region map and the diffused region vector field respectively.

where Δx and Δy are the spatial sample intervals, p_{max} is the maximum of $p(.)$, and Δt is time step, the interval between time t_n and time t_{n+1} when iteratively solving (1.12).

From (1.11) and (1.12) we note that within a homogeneous region, based on the criteria of region segmentation, $p(.)$ equals one while $q(.)$ equals zero. Thus (1.12) is only left with the first term (as the second term vanishes). This effectively smooths the vector field. However, at the region boundaries, $p(.) \to 0$ and $q(.) \to 1$. The smoothing term imposes less and the region vectors are close to the gradient of the region map R. Thus the diffused region vector field provides the evolving snake with an attracting force in a sufficiently large range near the region boundaries, and also allows the snake to evolve solely under other gradient forces.

Figure 1.12 illustrates an example of region force diffusion, including close-up views of pre and post-diffusion vector field.

1.4.3 Region-aided snake formulation

Next, we can derive the region-aided geometric snake formulation. The standard geometric snake is given by (1.8). In the traditional sense, the snake forces fall into two types, internal forces and external forces. The internal forces impose regularity on the curve and control the elasticity and rigidity of the snake. The external forces pull the snake towards salient image features such as object

boundaries. Thus, the internal and external forces in (1.8) can be written as

$$\begin{cases} F_{int} = g(|\nabla I|)\kappa\vec{\mathcal{N}} \\ F_{ext} = g(|\nabla I|)c\vec{\mathcal{N}} - \nabla g(|\nabla I|) \end{cases} \tag{1.14}$$

where $g(.)$ is the stopping function as before. The first term of the external forces is a constant shrink or expand force in the normal direction of the snake. It can be separated from other external forces in the sense that it is not spatially static in the image domain as other external forces and needs different numerical schemes. However, considering the previous definition of snake forces and that the constant force alone can push the snake towards boundaries, we keep it in the external term.

The diffused region force is a feature driven force and spatially static. So we can add the diffused region force to the external term:

$$\begin{cases} F_{int} = g(|\nabla I|)\kappa\vec{\mathcal{N}} \\ F_{ext} = \alpha g(|\nabla I|)\vec{\mathcal{N}} + \beta\tilde{R} - \nabla g(|\nabla I|) \end{cases} \tag{1.15}$$

where $\tilde{R}$ is the region force vector field obtained in (1.10) and α is a new constant incorporating c. Constants α and β act as a trade-off between gradient forces and region forces. In practice, β is a constant from 0 to 1 for most non-highly textured images. If good segmentation results are available, β should be set close to 1.

The snake evolves under all the internal and external forces. However, only the forces in the normal direction of the evolving contours can change the geometry. The forces tangential to the contours can only change the parameterisation of the contours. Thus, a geometric snake evolving under internal and external forces can be interpolated as

$$C_t = [(F_{int} + F_{ext}) \cdot \vec{\mathcal{N}}]\vec{\mathcal{N}} \tag{1.16}$$

Finally, by substituting (1.15) into (1.16), the region-aided geometric snake formulation becomes:

$$C_t = [g(|\nabla I|)(\kappa + \alpha) - \nabla g(|\nabla I|) \cdot \vec{\mathcal{N}} + \beta\tilde{R} \cdot \vec{\mathcal{N}}]\vec{\mathcal{N}} \tag{1.17}$$

1.4.4 Level set representation

In this section, we outline the level set representation for the region-aided geometric snake. Level sets describe a moving front in an implicit function and are the basis for the numerical algorithm for curve evolution according to functions of curvature, introduced by Osher and Sethian [15, 16]. In the application to active contours, the evolving contour is embedded into a higher dimensional surface as a zero level set. The entire surface, the level sets, is an implicit representation of the embedded contour. As shown in Figure 1.13, the snake is initially built in a three dimensional surface, which later evolves according to underlying forces. Finally, the converged snake is extracted from the level sets by cutting it at zero height.

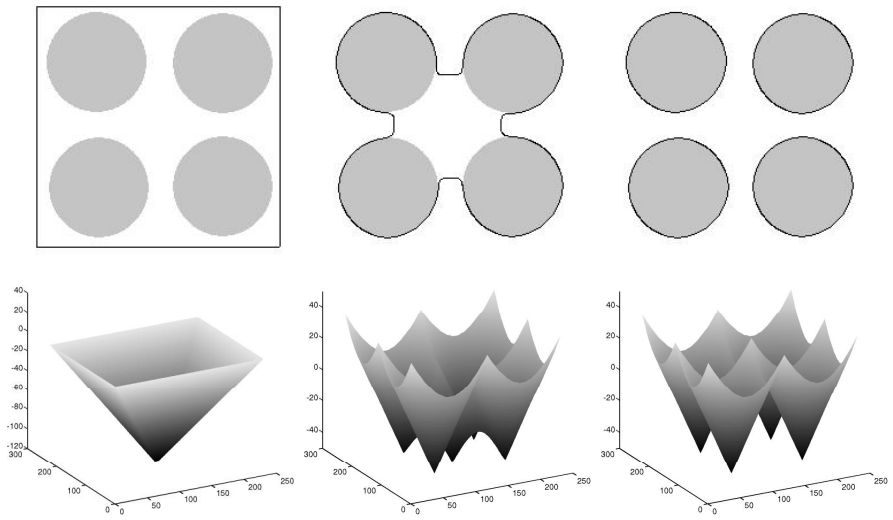

Figure 1.13: Level sets evolution for an embedded snake - top row: initial snake on test image, evolving contour, and final converged snake - bottom row: Corresponding evolving level sets. The snake is tracked at zero height.

Let C be a level set of a function of $\phi : [0, a] \times [0, b] \rightarrow \Re$. That is, C is embedded into the zero level set with ϕ an implicit representation of the curve C. This representation is parameter free and intrinsic. Given a planar curve that evolves according to $C_t = \mathbb{F}\vec{\mathcal{N}}$ for a given function $\mathbb{F}$, then the embedding function should deform according to $\phi_t = \mathbb{F}|\nabla\phi|$, where $\mathbb{F}$ is computed on the level sets. By embedding the evolution of C in that of ϕ, topological changes of C are handled automatically and, accuracy and stability are achieved using numerically stable computations.

The internal curvature and external pressure terms of the RAGS formulation in (1.17) can be easily transferred to a level set representation:

$$\begin{cases} C_t = g(|\nabla I|)\kappa\vec{\mathcal{N}} \rightarrow \phi_t = g(|\nabla I|)\kappa|\nabla\phi| \\ C_t = g(|\nabla I|)\alpha\vec{\mathcal{N}} \rightarrow \phi_t = g(|\nabla I|)\alpha|\nabla\phi| \end{cases} \tag{1.18}$$

The other external forces in (1.17) are static vector fields derived from image data which do not change as the active contour deforms. Static force fields are defined on the spatial positions rather than the active contour itself. Since $\vec{\mathcal{N}}$ is the inward normal, the level set representation of the inward unit normal is given by

$$\vec{\mathcal{N}} = -\frac{\nabla\phi}{|\nabla\phi|} \tag{1.19}$$

Then, we have

$$\mathbb{F} \cdot \vec{\mathcal{N}} = -\frac{1}{|\nabla\phi|}(\mathbb{F} \cdot \nabla\phi) \tag{1.20}$$

Combining (1.18) with (1.20) where $\mathbb{F}$ takes on the static force fields, the level set representation of RAGS is given by:

$$\phi_t = g(|\nabla I|)(\kappa + \alpha)|\nabla\phi| + \nabla g(|\nabla I|) \cdot \nabla\phi - \beta\tilde{R} \cdot \nabla\phi \tag{1.21}$$

where $g(.)$ is the stopping function as before. The expression for the curvature of the zero level set assigned to the interface itself is given by

$$\kappa = \mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) = \frac{\phi_{xx}\phi_y^2 - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{3/2}} \tag{1.22}$$

1.5 Numerical Solutions

The numerical solution for region force diffusion is discussed in detail in 1.5.1, but the detailed numerical solutions for RAGS level set representation are only presented in Appendix A as they are not critical to understanding the concepts underlying RAGS. In fact, the whole of this section can be skipped without loss of continuity.

1.5.1 Numerical solutions for region force diffusion for RAGS

Initially, a mesh grid needs to be selected, with final accuracy directly dependent on its resolution. However, due to the nature of a digital image, the grid resolution is constrained to the pixel level. It was shown in Section 1.4.2 that the steady solution of (1.10) can be achieved by computing the equilibrium state of (1.12). The initial state of the region force vector field $\tilde{R}$ is given by the gradient of the region map R. Simple central differences can be used to approximate ∇R, resulting in vectors that are then diffused. Let Δx and Δy be the grid spacing, Δt be the time step, and i, j, and n represent the spatial position and time. The partial derivative of time can be approximated by forward difference as

$$u_t = \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} \tag{1.23}$$

The spatial partial derivatives can be solved using central differences approximation given by

$$\nabla^2 u = \frac{u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j}}{\Delta x \Delta y} \tag{1.24}$$

The solutions to partial derivatives of $v(x, y, t)$ are similar to those of $u(x, y, t)$. The weighting functions given in (1.11) can be easily computed. Thus, substituting the partial derivatives into (1.12) gives the following iterative solution:

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t\Lambda \\ v_{i,j}^{n+1} = v_{i,j}^{n} + \Delta t\Omega \end{cases} \tag{1.25}$$

where,

$$\Lambda = \frac{p(.)_{i,j}}{\Delta x \Delta y}(u^n_{i+1,j} + u^n_{i,j+1} + u^n_{i,j-1} + u^n_{i-1,j} - 4u^n_{i,j}) - q(.)_{i,j}(u^n_{i,j} - R_{x,ij})$$

and

$$\Omega = \frac{p(.)_{i,j}}{\Delta x \Delta y}(v^n_{i+1,j} + v^n_{i,j+1} + v^n_{i,j-1} + v^n_{i-1,j} - 4v^n_{i,j}) - q(.)_{i,j}(v^n_{i,j} - R_{y,ij})]$$

where $R_{x,ij}$ and $R_{y,ij}$ are partial derivatives of R. They also can be approximated by central differences as

$$\begin{cases} R_{x,ij} = \frac{R_{i+1,j} - R_{i-1,j}}{2\Delta x} \\ R_{y,ij} = \frac{R_{i,j+1} - R_{i,j-1}}{2\Delta y} \end{cases} \tag{1.26}$$

The convergence is guaranteed with the time step restriction of (1.13).

1.5.2 Numerical solution for the level set implementation of RAGS

As in the numerical solution for vector diffusion, a computational grid is required. Once the grid is chosen, the initial level sets $\phi(x,t) = 0$ can be defined with the property that the zero level set corresponds to the initial contours of the snake. The signed-distance transform can be used to build the initial level sets. A brute-force Euclidean distance transform would be computationally infeasible. Practically, accuracy is required only near the initial contours and discrete values based on grid distance can suffice further away. A positive sign is given to the points outside the contours, and a negative sign is applied to the points inside.

As shown in (1.17), the snake evolves according to four forces that can be catagorised into three types based on the nature of their motions:

1. The first motion is of a smoothing and collapsing nature with speed proportional to its curvature as shown in Figure 1.1. It can be numerically approximated using central differences, because the curvature is only dependent on the contour. It is independent of time and spatial position.

2. The second is expanding or shrinking with a spatially constant speed, characterised by $\alpha g(.)$ in the normal direction of the curve. However, when the constant term exists the normals can collide with each other while evolving. Thus shocks, or corners, will form and once a shock has developed, some information will be lost as it evolves. This means that shocks cause irreversibility; information can not be recovered by tracing 'backwards' in time. Generally, no new information can be created while evolving, which is referred to as an *entropy condition*. Central difference approximation can not be used to approximate the gradient in this case, as it suffers from shocks where the entropy condition is invoked. An *upwind scheme* can be

used, as an entropy-satisfying scheme, that engages information upwind of the direction of its propagation. In other words, in order to achieve a stable numerical scheme, the numerical domain of dependence should contain the mathematical domain of dependence. Thus, in order to approximate the gradient of the constant term, it is important first to know which way the speed function points, and whether it is negative or positive. Then we can choose proper backward or forward difference approximations.

3. The third type of motion in (1.17) is contributed by the underlying static velocity field, the direction and strength of which are based on spatial position. It is independent of the shape and position of the snake. The motion of contours under this velocity field can be numerically approximated through *upwind schemes* by checking the sign of each component of the velocity field and constructing one-sided upwind differences in the appropriate direction. For a positive speed component, backward difference approximation is used, otherwise forward difference approximation should be applied.

By using these approximation schemes, (1.17) can be numerically implemented. The detailed numerical solutions for RAGS are presented in Appendix A. For general numerical solution to level sets, including concepts such as *entropy condition* and *upwind scheme*, the interested reader is referred to works by Sethian [17, 16] and by Osher and Fedkiw [18].

1.6 Region-aided Geometric Snake on Vector-Valued Images

The theory of boundary detection by the geometric or geodesic snake can be applied to any general 'edge detector' function. The stopping function g should tend to zero when reaching edges.

When dealing with greylevel images, the decreasing function g can be easily derived from the edge detector f, as shown in (1.9). We use a similar stopping function for edges obtained directly from vector-valued images such as a colour image.

A consistent extension of scalar gradients based on a solid theoretical foundation has been presented by di Zenzo [19]. This extension has been applied in the active contour literature to both geometric and parametric snakes.

In a vector-valued image the vector edge is considered as the largest difference between eigenvalues in the tensor metric. Let $\Theta(u_1, u_2) : \Re^2 \rightarrow \Re^m$ be a m-band image for $i = 1, 2, ..., m$. For colour images, $m = 3$. A point in the image is considered as a vector in $\Re^m$. The distance between two points, $P = (u_1^0, u_2^0)$ and $Q = (u_1^1, u_2^1)$, is given by $\triangle\Theta = \Theta(P) - \Theta(Q)$. When this distance tends to the infinitesimal, the difference becomes the differential $d\Theta = \sum_{i=1}^{2} \frac{\partial\Theta}{\partial u_i} du_i$

with its squared norm given by

$$d\Theta^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{\partial\Theta}{\partial u_i}\frac{\partial\Theta}{\partial u_j} du_i du_j \tag{1.27}$$

Using standard Riemannian geometry notation, then let $s_{ij} = \frac{\partial\Theta}{\partial u_i} \cdot \frac{\partial\Theta}{\partial u_j}$, such that

$$d\Theta^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} s_{ij} du_i du_j = \begin{bmatrix} du_1 \\ du_2 \end{bmatrix}^T \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} du_1 \\ du_2 \end{bmatrix} \tag{1.28}$$

For a unit vector $v = (\cos\theta, \sin\theta)$, then $d\Theta^2(v)$ indicates the rate of change of the image in the direction of v. The extrema of the quadratic form are obtained in the directions of the eigenvectors of the metric tensor s_{ij}, and the corresponding eigenvalues are:

$$\lambda_{\pm} = \frac{s_{11} + s_{22} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}}{2} \tag{1.29}$$

with eigenvectors $(\cos\theta_{\pm}, \sin\theta_{\pm})$ where the angles $\theta_{\pm}$ are given by

$$\begin{cases} \theta_+ = \frac{1}{2}\arctan\frac{2s_{12}}{s_{11}-s_{22}} \\ \theta_- = \theta_+ + \frac{\pi}{2} \end{cases} \tag{1.30}$$

The maximal and minimal rates of change are the λ_+ and λ_- eigenvalues respectively, with corresponding directions of change θ_+ and θ_-. The strength of an edge in a vector-valued case is not given simply by the rate of maximal change λ_+, but by the difference between the extremums. Hence, a good approximation function for the vector edge magnitude should be based on $f = f(\lambda_+, \lambda_-)$. Now RAGS can be extended to the region-aided geometric colour snake by selecting an appropriate edge function f_{col}. The edge stopping function g_{col} is defined such that it tends to 0 as $f_{col} \to \infty$. The following functions can be used (cf. (1.9)):

$$f_{col} = \lambda_+ - \lambda_- \quad \text{and} \quad g_{col} = \frac{1}{1 + f_{col}} \tag{1.31}$$

Then replacing $g_{col}(.)$ for the edge stopping term $g(.)$ in (1.17), we have the colour RAGS snake:

$$C_t = [g_{col}(|\nabla I|)(\kappa + \alpha) - \nabla g_{col}(|\nabla I|) \cdot \vec{\mathcal{N}} + \beta\tilde{R} \cdot \vec{\mathcal{N}}]\vec{\mathcal{N}}. \tag{1.32}$$

Finally, its level set representation is also given by replacing $g_{col}(.)$ for $g(.)$ in (1.21):

$$\phi_t = g_{col}(|\nabla I|)(\kappa + \alpha)|\nabla\phi| + \nabla g_{col}(|\nabla I|) \cdot \nabla\phi - \beta\tilde{R} \cdot \nabla\phi \tag{1.33}$$

1.7 The Mean Shift Algorithm

This section can be skipped without loss of continuity. Its topic is the process of generating the image region segmentation map S which is then used as described in Section 1.4.2. The reader can assume it is available and skip to the next section.

An essential requisite for RAGS is a segmentation map of the image. This means that RAGS is independent of any particular segmentation technique as long as a region map is produced, however, it is dependent on its representational quality. In this section, the mean shift algorithm is reviewed as a robust feature space analysis method which is then applied to image segmentation. It provides very reasonable segmentation maps and has extremely few parameters that require tuning.

The concept underlying the nonparametric mean shift technique is to analyse the density of a feature space generated from some input data. It aims to delineate dense regions in the feature space by determining the modes of the unknown density, i.e. first the data is represented by local maxima of an empirical probability density function in the feature space and then its modes are sought. The denser regions are regarded as significant clusters. Comaniciu and Meer [20, 13] have recently provided a detailed analysis of the mean shift approach, including the review below, and presented several applications of it in computer vision, e.g. for colour image segmentation.

We now briefly present the process of density gradient estimation. Consider a set of n data points $\{x_i\}_{i=1\ldots n}$ in the d-dimensional Euclidean space R^d. Also consider the Epanechnikov kernel, an optimum kernel yielding minimum mean integrated square error:

$$K(x) = \begin{cases} \frac{1}{2Z_d}(d+2)(1-x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.34}$$

where Z_d is the volume of the unit d-dimensional sphere. Using $K(x)$ and window radius h, the *multivariate kernel density estimate* on the point x is

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big). \tag{1.35}$$

The estimate of the density gradient can be defined as the gradient of the kernel density estimate since a differentiable kernel is used:

$$\hat{\nabla} f(x) \equiv \nabla \hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \nabla K\Big(\frac{x - x_i}{h}\Big). \tag{1.36}$$

Applying (1.34) to (1.36), we obtain:

$$\hat{\nabla} f(x) = \frac{n_x}{n(h^d Z_d)} \frac{d+2}{h^2} \Big(\frac{1}{n_x} \sum_{x_i \in H_h(x)} [x_i - x]\Big) \tag{1.37}$$

where the region $H_h(x)$ is a hypersphere of radius h and volume $h^d Z_d$, centered on x, and containing n_x data points. The *sample mean shift* is the last term in (1.37)

$$M_h(x) \equiv \frac{1}{n_x} \sum_{x_i \in H_h(x)} [x_i - x] \tag{1.38}$$

The quantity $\frac{n_x}{n(h^d Z_d)}$ is the kernel density estimate $\hat{f}(x)$ computed with the hypersphere $H_h(x)$, and thus (1.37) can be rewritten as

$$\hat{\nabla} f(x) = \hat{f}(x) \frac{d+2}{h^2} M_h(x), \tag{1.39}$$

which can be rearranged as

$$M_h(x) = \frac{h^2}{d+2} \frac{\hat{\nabla} f(x)}{\hat{f}(x)}. \tag{1.40}$$

Using (1.40), the mean shift vector provides the direction of the gradient of the density estimate at x which always points towards the direction of the maximum increase (in the density). Hence, it converges along a path leading to a mode of the density.

In [13], Comaniciu and Meer performed the mean shift procedure for image segmentation in a joint domain, the image (*spatial*) domain and colour space (*range*) domain. The spatial constraints were then inherent in the mode searching procedure. The window radius is the only significant parameter in their segmentation scheme. A small window radius results in over-segmentation (i.e. larger number of clusters), and a large radius produces under-segmentation (yielding a smaller number of clusters). In this work, the performance of RAGS will be demonstrated on both the under-segmentation and over-segmentation resolutions of Comaniciu and Meer's work. In either case, the result of the mean shift procedure is the region segmentation map S which is passed to RAGS for generating the diffused region boundary map $\tilde{R}$.

1.8 A summary of the RAGS algorithm

The colour RAGS algorithm is now reviewed with the aid of Figure 1.14. Given the input colour image, two streams of processing can begin concurrently.

- In the first stream, the vector gradient is computed to provide the edge function f, which is then used in (1.9) to yield the decreasing function g, followed by ∇g. Function g will act as spatial weights for the snake curvature force and constant force, and ∇g will contribute to the underlying doublet attraction force.

- In the second stream, a region segmentation map S is produced by applying any reasonable segmentation technique, e.g. the mean shift algorithm. From it, then region map R can be generated using vector gradients.

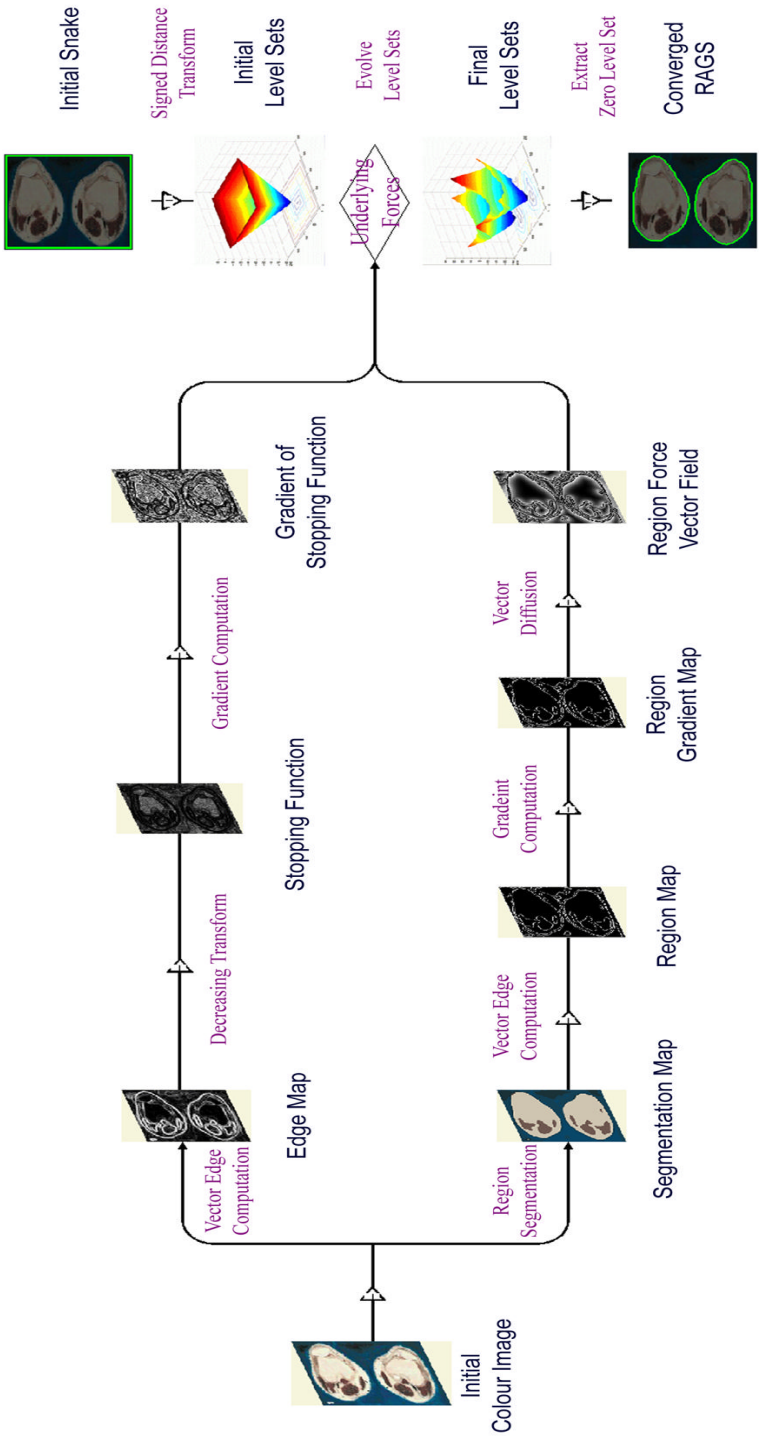

Figure 1.14: RAGS processing schema. (Colour Slide)

Gradient of the region map R provides ∇R, which imposes region forces immediate to region boundaries. These region forces are then diffused by solving (1.10), resulting in a region force vector field $\tilde{R}$.

Thus, all the underlying velocity fields and the weighting function g are ready and prepared. Then we can generate initial level sets based on an initial snake using the distance transform and evolve the level sets according to all force fields (rightmost part of Figure 1.14). The curvature force and constant force adaptively change with the level set snake. Along with the static forces, they are numerically solved using the principles described in section 1.5.2 with the solutions given in Appendix A. After the level set evolves to a steady state, the final snake is easily obtained by extracting the zero level set.

1.9 Experiments and Results

In this section we present results that show improvements over either the standard geometric snake or the geometric GGVF snake or both, and mainly in images where there are weak edges or noisy regions preventing the aforementioned snakes to perform at their best. Although GGVFs have been reported only using greylevel image gradients, we can also apply them to 'colour' gradients (obtained as described in section 1.6), which allows direct comparison with the colour RAGS. It must also be noted that the GGVF can sometimes perform better than we have shown in some of the following examples as long as it is initialised differently, i.e much closer to the desired boundary. In all the experiments, we have initiated the geometric, GGVF, and RAGS snakes at the same starting position, unless specifically stated.

1.9.1 Preventing Weak-edge Leakage

We first illustrate the way weak-edge leakage is handled on a synthetic image. The test object is a circular shape with a small blurred area on the upper right boundary as shown in Figure 1.15.

The standard geometric snake steps through the weak edge because the intensity changes so gradually that there is no clear boundary indication in the edge map. The RAGS snake converges to the correct boundary since the extra diffused region force delivers useful global information about the object boundary and helps prevent the snake from stepping through. Figure 1.16 shows, for the test object in Figure 1.15, the edge map, the stopping function $g(.)$, its gradient magnitude $|\nabla g(.)|$, the region segmentation map S, and the vector map of the diffused region force $\tilde{R}$.

1.9.2 Neighbouring Weak/Strong Edges

The next experiment is designed to demonstrate that both the standard geometric snake and the GGVF snake readily step through a weak edge to reach a neighbouring strong edge. The test object in Figure 1.17 contains a prominent

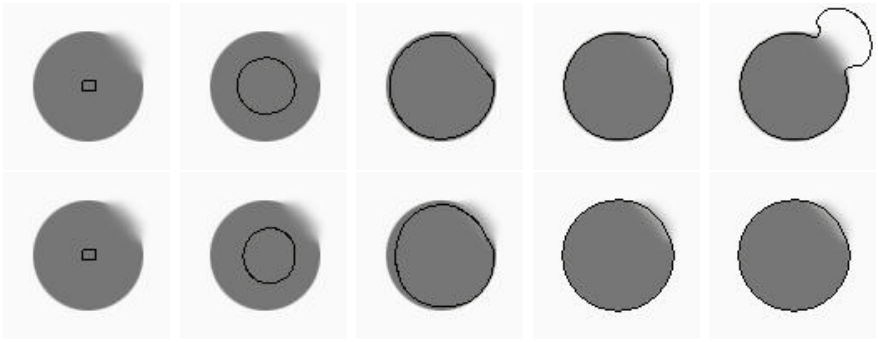

Figure 1.15: Weak-edge leakage testing on a synthetic image - top row: geodesic snake steps through - bottom row: RAGS snake converges properly using its extra region force.

Figure 1.16: Diffused region force on weak edge - from left: the edge map, the stopping function $g(.)$ of edge map, the magnitude of its gradient $\nabla g(.)$, the region segmentation map, and the vector map of the diffused region force $\tilde{R}$.

circle inside a faint one. The presence of the weaker edge at the outer boundary is detected only by the RAGS snake. The geodesic snake fails because the weaker outer boundary allows the whole snake to leak through (similar to but in the opposite direction of propagation in Figure 1.15). The GGVF snake fails due to the strong gradient vector force caused by the inner object boundary. Practical examples of this can also be observed in most of the real images shown later, such as Figures 1.20 and 1.26.

1.9.3 Testing on Noisy Images

We also performed comparative tests to examine and quantify the tolerance to noise for the standard geometric, the geometric GGVF, and the RAGS snakes. For this a Harmonic shape was used as shown in Figure 1.18. It was generated using

$$r = a + bcos(m\theta + c) \tag{1.41}$$

where r is the length from any edge point to the centre of the shape, a, b, and c remain constant and m can be used to produce different numbers of 'bumps'; in this case $m = 6$. We added varying amounts of noise and measured the accuracy of fit (i.e. boundary description) after convergence. The accuracy was

Figure 1.17: Strong neighbouring edge leakage - from left: initial snake, geodesic snake steps through weak edge in top right of outer boundary, GGVF is attracted by the stronger inner edge, RAGS snake converges properly using extra region force.

computed using Maximum Radial Error (MRE), i.e. the maximum distance in the radial direction between the true boundary and each active contour.

Figure 1.18: A shape and its boundary (a Harmonic curve).

Impulse noise was added to the original image from 10% to 60% as shown in the first column of Fig 1.19. The region segmentation data used for RAGS is in the second column (without any post-processing to close gaps etc.). The third, fourth, and fifth columns show the converged snake for the standard geometric, the GGVF, and RAGS snakes respectively. A simple subjective examination clearly demonstrates the superior segmentation quality of the proposed snake. The initial state for the standard geometric and RAGS snakes is a square at the edge of the image, while for the GGVF it is set close to the true boundary to ensure better convergence. At low percentages of noise, all snakes could find the boundary accurately enough. However, at increasing noise levels ($> 20\%$), more and more local maximums appear in the gradient flow force field, which prevent the standard geometric and GGVF snakes from converging to the true boundaries. The RAGS snake has a global view of the noisy image and the underlying region force pushes it towards the boundary. The MRE results are shown in Table 1.1. These verify RAGS error values to be consistently and significantly lower than the other two snake types for noise levels $> 10\%$.

1.9.4 Results on greylevel images

Figures 1.20 to 1.22 demonstrate RAGS in comparison to the standard geometric and GGVF snakes on various greylevel images. Figure 1.20 shows a good

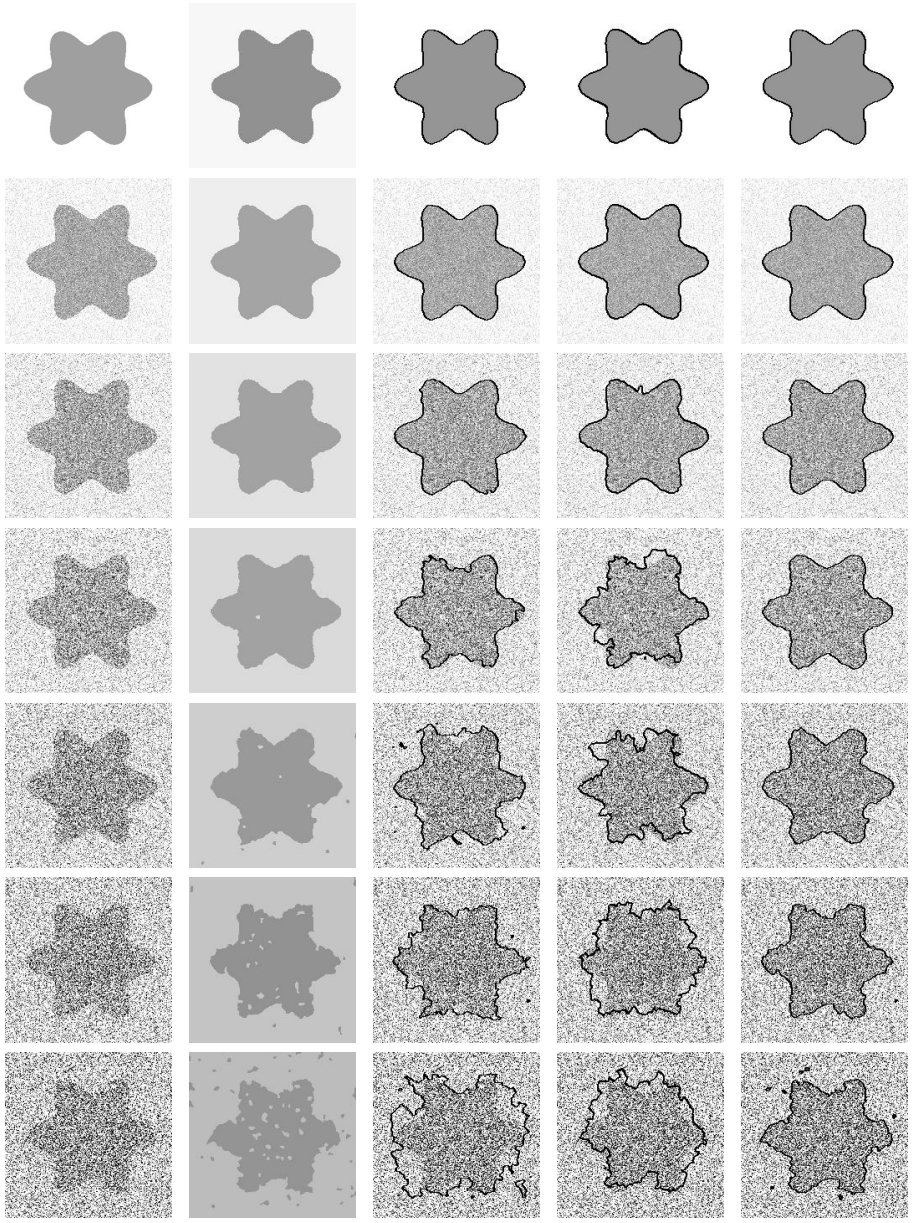

Figure 1.19: Shape recovery in noisy images - (column 1) original image with various levels of added Gaussian noise [0%,10%,...,60%], (column 2) the region maps later diffused by RAGS, (column 3) standard geometric snake results, (column 4) GGVF snake results, (column 5) RAGS results.

Table 1.1: MRE comparison for the Harmonic shapes in Figure 1.19.

% noise	Stand. Geom. Snake Error	GGVF Snake Error	RAGS Snake Error
0	2.00	2.00	**2.00**
10	2.23	2.24	**2.00**
20	5.00	7.07	**4.03**
30	10.00	16.03	**3.41**
40	16.16	21.31	**5.22**
50	15.81	21.00	**5.38**
60	28.17	20.10	**5.83**

example of weak-edge leakage on the lower side of the object of interest. While RAGS does extremely well here, the geometric snake leaks through and the GGVF snake leaks and fails to progress at all in the narrow object. In Figure 1.21, RAGS achieves a much better overall fit than the other snakes, particularly in the lower regions of the right-hand snake and the upper-right regions of the left-hand snake. In Figure 1.22, again RAGS manages to segment the desired region much better than the standard geometric and the GGVF snakes. Note the standard snake leaks out of the object, similar to the effect demonstrated with the synthetic image in Figure 1.15.

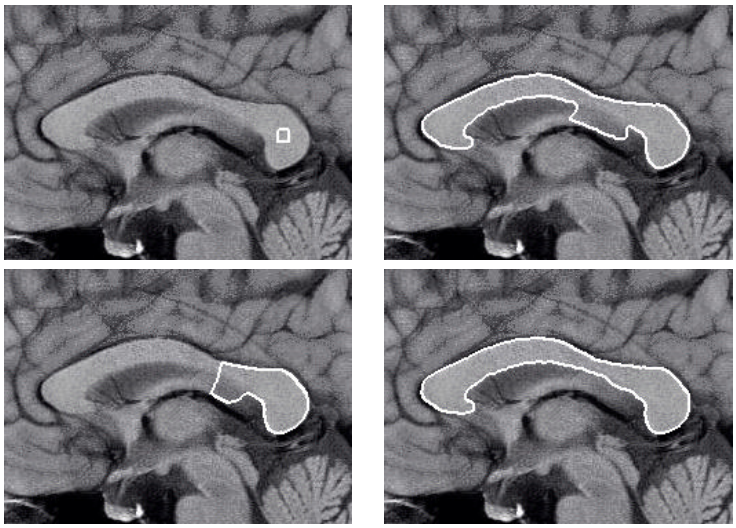

Figure 1.20: Brain MRI (corpus callosum) image - top row: initial snake, standard geometric snake - bottom row: GGVF snake, and RAGS snake (*original image courtesy of GE Medical Systems*).

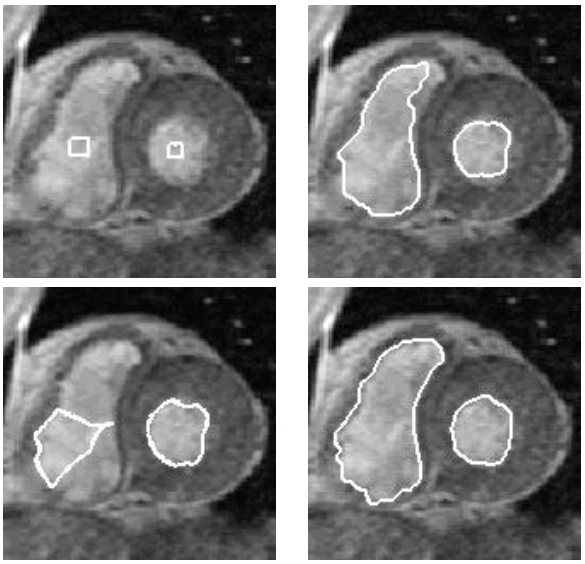

Figure 1.21: Heart MRI image - top row: initial snakes, standard geometric snakes - bottom row: GGVF snakes, and final RAGS snakes showing improvement on the top right of the left snake and the lower region of the right snake.

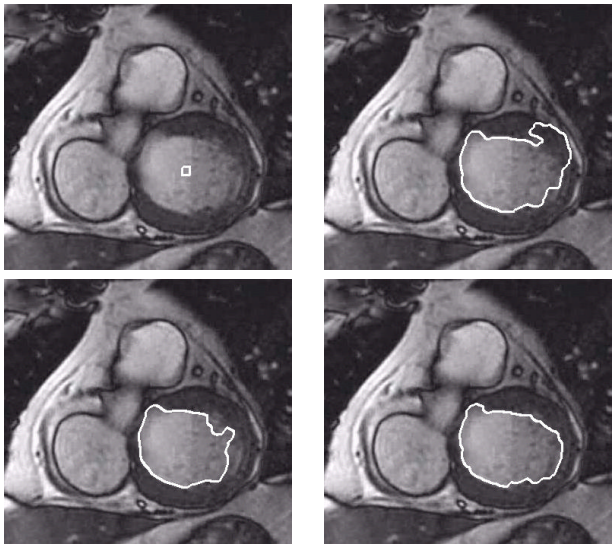

Figure 1.22: Heart MRI image - top row: initial snake, standard geometric snake - bottom row: GGVF snake, and final RAGS snake showing better convergence and no leakage. (*original image courtesy of GE Medical Systems*).

1.9.5 Results on colour images

We now consider the performance of the RAGS snake on colour images. In Figure 1.23 we can see a cell image with both strong and fuzzy region boundaries. Note how the fuzzy boundaries to the right of the cell 'dilute' gradually into the background. So the results in the top-right image again demonstrate an example of weak edge leakage, similar to the example in Figure 1.22, where the standard geometric snake fails to converge on the outer boundary. The middle and bottom rows show the converged RAGS snake using the over-segmentation and under-segmentation colour region maps produced by the mean shift algorithm.

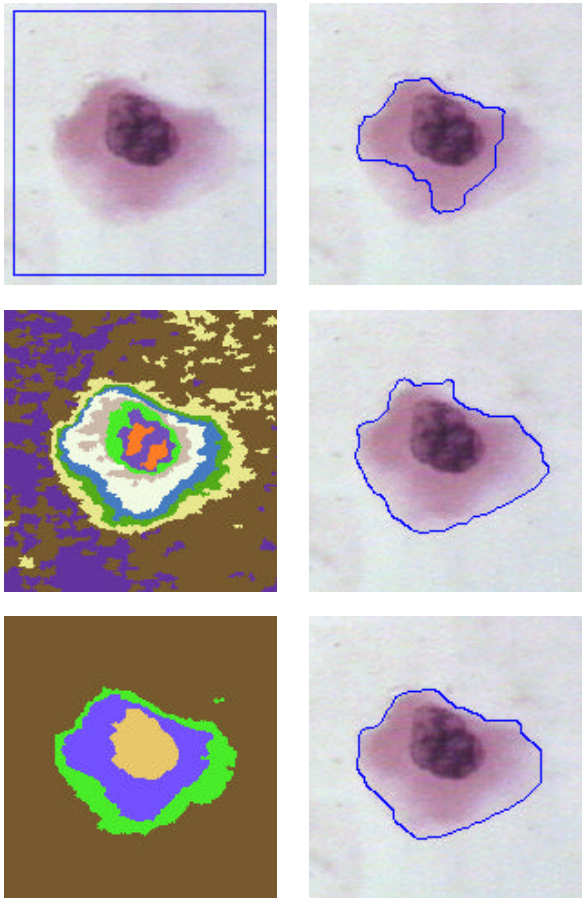

Figure 1.23: Weak-edge leakage testing - top row: original image with starting contour and geodesic snake which steps through - middle row: over-segmentation colour region map and converged RAGS snake - bottom row: under-segmentation colour region map and converged RAGS snake (*Original image courtesy of Bristol Biomedical Image Archive, Bristol University, UK*). (Colour Slide)

A very similar example is demonstrated in Figure 1.24 in application to images of the optic disk in which the blood vessels have been removed using colour mathematical morphology techniques. Again, the failing performance of the standard snake is shown along with the RAGS results on both over-segmentation and under-segmentation regions.

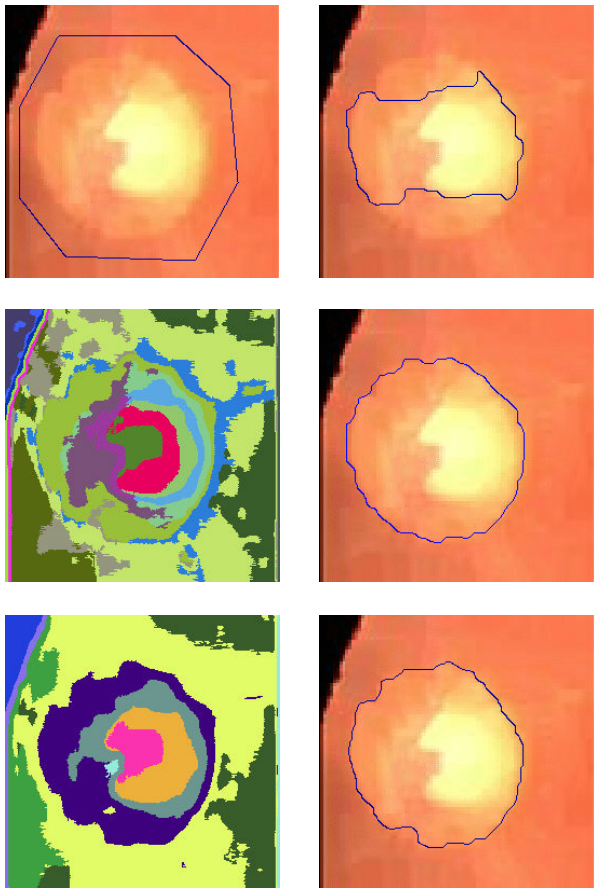

Figure 1.24: Optic disk localisation - top row: initial contour and geodesic snake which steps through to the stronger central region - middle row: over-segmentation colour region map and final RAGS snake - bottom row: under-segmentation colour region map and final RAGS snake. (Colour Slide)

In Figure 1.25, a full application of RAGS is presented where the resulting regions from the RAGS snake are quantitatively evaluated against those hand-labelled by an expert ophthalmologist. The first column represents these groundtruth boundaries. The second column shows the position of the starting RAGS snakes. The boundary of the optic disk is quite fuzzy and well blended with the background. The region force helps the proposed snake stop at weak

edges while the standard geometric snake leaks through (as shown in Figure 1.24) and the accuracy of the GGVF snake is highly dependent on where the initial snake is placed (hence GGVF snake results are not provided). The last two columns illustrate the RAGS results using over-segmented and under-segmented regions of the mean shift algorithm respectively.

A simple measure of overlap is used to evaluate the performance of the RAGS snake against its corresponding groundtruth:

$$M = \frac{n(A \cap B)}{n(A \cup B)}$$

where A and B correspond to ground-truth and RAGS localised optic disk regions respectively, and $n(.)$ is the number of pixels in a region. Table 1.2 shows the result of measurement M demonstrating a 91.7% average performances for both over/under-segmentation RAGS respectively.

Table 1.2: Quantitative evaluation of RAGS snake on the optic disks in 1.25.

Image	1	2	3	4	5	6	**Avg.**
% RAGS(over)	91.4	90.0	91.9	93.1	93.1	90.5	**91.7**
% RAGS(under)	90.7	89.5	93.1	91.3	93.0	92.7	**91.7**

The final example in Figure 1.26 shows a darker cell centre compared to the cell outer region, but more significantly the object of interest is surrounded by other strong features. The standard geometric snake splits and converges unsatisfactorily and the GGVF snake is pulled in and out by the stronger inner cell nucleus and neighbouring cells respectively, while the RAGS snake converges well to the outer cell boundary without leaking through.

All the examples shown here illustrate the resilience of RAGS to weak edges and noise. However, the RAGS snake does suffer some shortcomings. As with the standard geometric snake, or the geometric GGVF snake, it will not perform well in highly textured regions in which the gradient flow forces may be hampered by multitudes of texture edge information. It is also dependent on a reasonable segmentation stage, although this was shown to be quite flexible using a popular method of image segmentation.

1.10 Conclusions

A novel method, the region-aided geometric snake or RAGS, was proposed. It integrates the gradient flow forces with region constraints, composed of the image region vector flow forces obtained through the diffusion of the region map. The theory behind RAGS is standalone and hence the region force can be generated starting from any reasonable segmentation technique. We also showed its simple extension to colour gradients. We demonstrated the performance of RAGS, against the standard geometric snake and the geometric GGVF snake, on weak edges and noisy images as well as on a number of other examples.

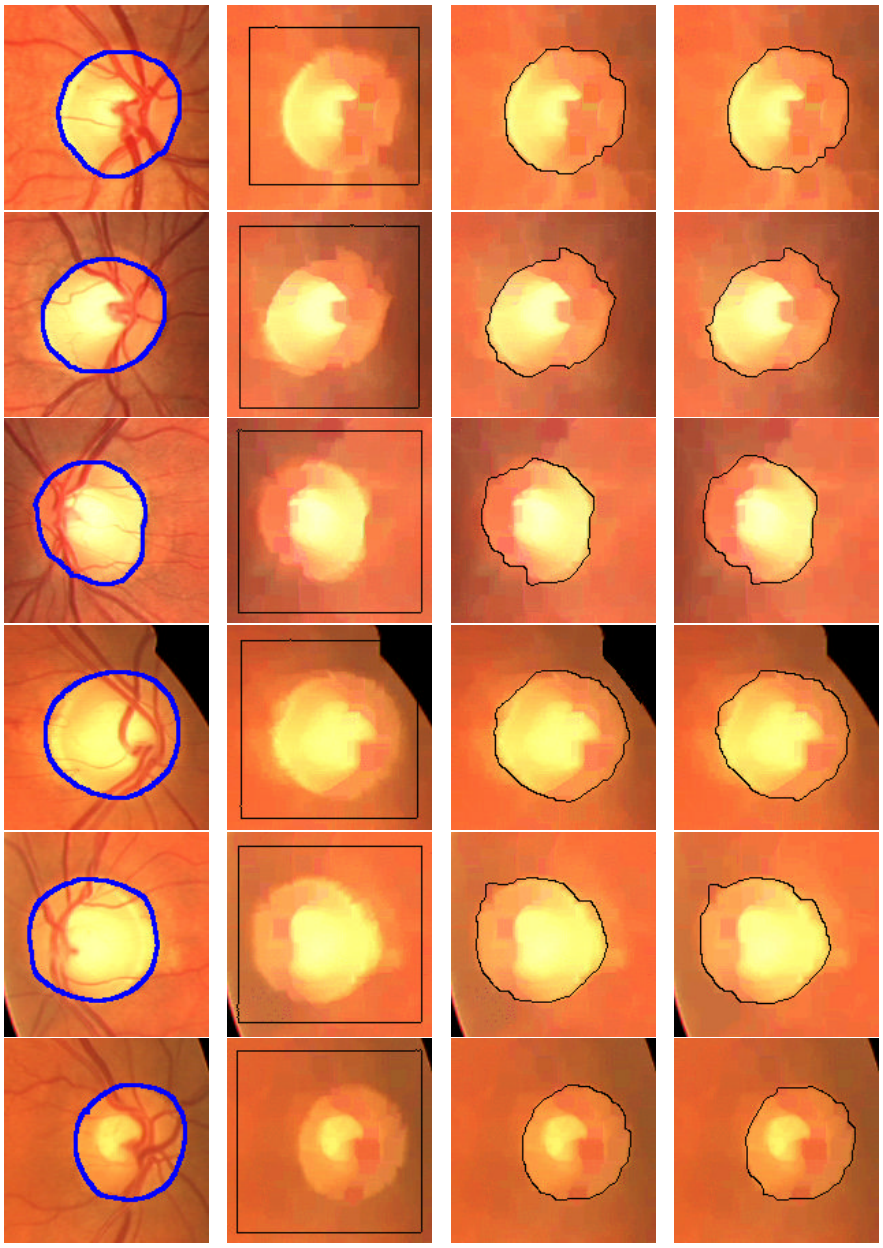

Figure 1.25: RAGS segmentation comparison with ground-truth - (column 1) ground-truth, (column 2) initial snakes, (column 3) RAGS results with over-segmentation, (column 4) RAGS results with under-segmentation (Colour Slide).

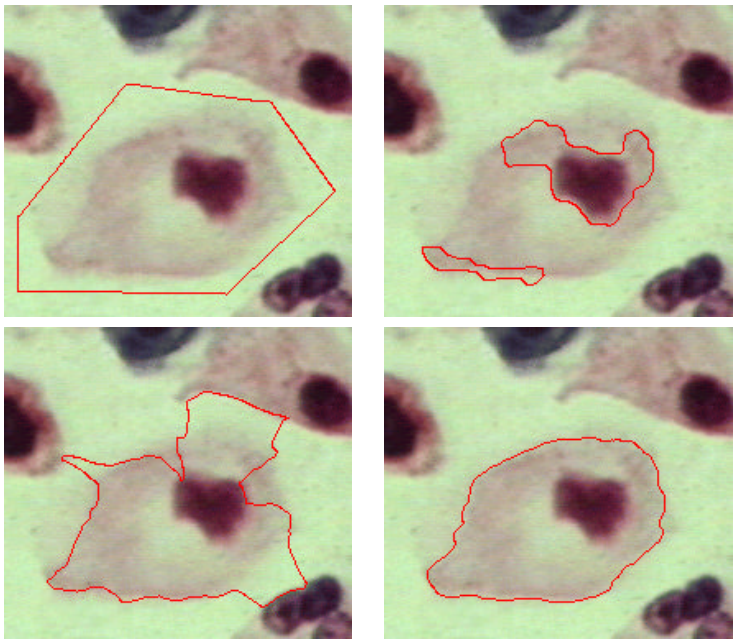

Figure 1.26: Cell with strong nucleus feature - top row: initial snake, standard geometric snake - bottom row: GGVF snake, RAGS snake showing how the stronger inner edge in the cell nucleus does not cause it to lose the outer weaker edge (*original image courtesy of Bristol Biomedical Image Archive, Bristol University, UK*). (Colour Slide)

The experimental results have shown that the region-aided snake is much more robust towards weak edges. Also it has better convergence quality compared with both the standard geometric snake and the geometric GGVF snake. The weak-edge leakage problem is usually caused by inconclusive edge values at the boundaries, which makes it difficult for gradient-based techniques to define a good edge. The gradual changes do not provide a sufficient minima for the stopping function to prevent the level set accumulating in that area. The diffused region map gives the snake an extra underlying force at the boundaries. It also makes the snake more tolerable to noise as shown by the harmonic shape recovery experiment and many of the real images. The noise in the image introduces local minima in the stopping function preventing the standard geometric snake to converge to the true boundary. However, for RAGS the diffused region forces give a better global idea of the object boundary in the noise clutter and helps the snake step closer and converge to the global minima.

1.11 Further Reading

Deformable contour models are commonly used in image processing and computer vision, for example for shape description [21], object localisation [22], and visual tracking [23].

A good starting point to learn about *parametric* active contours is [24]. These snakes have undergone significant improvements since their conception, for example see the GVF snake in [7, 9]. Region-based parametric snake frameworks have also been reported in [25, 26, 27].

The geometric model of active contours was simultaneously proposed by Caselles et al. [1] and Malladi et al. [2]. Geometric snakes are based on the theory of curve evolution in time according to intrinsic geometric measures of the image. They are numerically implemented via level sets, the theory of which can be sought in [15, 16].

There has been a number of works based on the geometric snake and level set framework. Siddiqi et al. [14] augmented the performance of the standard geometric snake that minimises a modified length functional by combining it with a weighted area functional. Xu and Prince extended their parametric GVF snake [7] into the Generalised GVF snake, the GGVF, in [9]. Later, they also established an equivalence model between parametric and geometric active contours [10] using the GGVF. A geometric GGVF snake enhanced with simple region-based information was presented in [10]. Paragios and Deriche [28, 29] presented a boundary and region unifying geometric snake framework which integrates a region segmentation technique with the geometric snake. In [30], Yezzi et al. developed coupled curve evolution equations and combined them with image statistics for images of a known number of region types, with every pixel contributing to the statistics of the regions inside and outside an evolving curve. Using colour edge gradients, Sapiro [6]) extended the standard geometric snake for use with colour images (also see Figure 1.6). In [11], Chan and Vese described a region-segmentation based active contour that does not use the geometric snake's gradient flow to halt the curve at object boundaries. Instead, this was modelled as an energy minimisation of a Mumford-Shah based minimal partition problem and implemented via level sets. Their use of a segmented region map is similar to the concept we have explored here.

Level set methods can be computationally expensive. A number of fast implementations for geometric snakes have been proposed. The *Narrow Band* technique, initially proposed by Chop [31], only deals with pixels that are close to the evolving zero level set to save computation. Later, Adalsterinsson and Sethian [32] analysed and optimised this approach. Sethian [33, 34] also proposed the fast marching method to reduce the computations, but it requires the contours to monotonically shrink or expand. Some effort has been expended in combining these two methods. In [35], Paragios et al. showed this combination could be efficient in application to motion tracking. Adaptive mesh techniques [36] can also be used to speed up the convergence of PDEs. More recently, additive operative splitting (AOS) schemes were introduced by Weickert et al. [37] as an unconditionally stable numerical scheme for nonlinear diffusion in image

processing. The basic idea is to decompose a multi-dimensional problem into one-dimensional ones. AOS schemes can be easily applied in implementing level set propagation [38].

The mean shift algorithm is a nonparametric technique for estimation of the density gradient, which was first proposed by Fukunaga and Hostetler [39]. The idea was later generalised by Cheng [40]. The technique was extended to various applications, amongst them colour image segmentation, by Comaniciu and Meer [20, 12, 13].

Appendix A

Numerical solution for the level set implementation of RAGS

Let $\phi_{i,j}$ denote the value of ϕ at the grid position of $x_i = i\Delta x, y_i = j\Delta y$, where Δx and Δy are the grid steps along x and y directions respectively. Denote $\phi(x_i, y_j, t_n)$ by $\phi^n_{i,j}$, the time derivative ϕ_t at (i, j, t_n) is approximated by forward difference as $\phi_t(i, j, t_n) = (\phi^{n+1}_{i,j} - \phi^n_{i,j})/\Delta t$, where Δt is a small time interval. As given in (1.17), the snake evolves according to four forces. However, they can be catagorised into three types based on the nature of their motions.

The first motion is a collapsing one with speed proportional to its curvature. It is a parabolic contribution to the equation of motion and it can be approximated with central differences. The curvature κ is only dependent on contours; it is independent of time and spatial position, hence it can also be solved using central difference approximations. The curvature motion at time t is approximated as

$$(g(.)\kappa|\nabla\phi|)^n_{i,j} = g(.)_{i,j}K^n_{i,j}(D^{0x}_{i,j}{}^2 + D^{0y}_{i,j}{}^2)^{1/2} \tag{1.42}$$

where $D^{0x}_{i,j} = (\phi^n_{i+1,j} - \phi^n_{i-1,j})/2\Delta x, D^{0y}_{i,j} = (\phi^n_{i,j+1} - \phi^n_{i,j-1})/2\Delta y$, and $K^n_{i,j}$ is the central difference approximation to the curvature expression given in (1.22):

$$(\phi_x)^n_{i,j} = D^{0x}_{i,j}, (\phi_y)^n_{i,j} = D^{0y}_{i,j} \tag{1.43}$$

$$(\phi_{xx})^n_{i,j} = \frac{\phi^n_{i+1,j} - 2\phi^n_{i,j} + \phi^n_{i-1,j}}{\Delta x^2}, (\phi_{yy})^n_{i,j} = \frac{\phi^n_{i,j+1} - 2\phi^n_{i,j} + \phi^n_{i,j-1}}{\Delta y^2} \tag{1.44}$$

$$(\phi_{xy})^n_{i,j} = \frac{\phi^n_{i+1,j+1} - \phi^n_{i-1,j+1} - \phi^n_{i+1,j-1} + \phi^n_{i-1,j-1}}{4\Delta x\Delta y} \tag{1.45}$$

The second motion is expanding or shrinking with a spatially constant speed in its normal direction. It must be approximated through *entropy-satisfying schemes* [16]. Let V_0 be the constant speed function regarding to $\alpha g(.)$. Following Sethian's *upwinding* finite difference scheme, the solution is given by

$$\begin{cases} (V_0|\phi|)^n_{i,j} = V_{0i,j}[max(D^{-x}_{i,j}, 0)^2 + min(D^{+x}_{i,j}, 0)^2 + \\ \qquad max(D^{-y}_{i,j}, 0)^2 + min(D^{+y}_{i,j})^2]^{1/2} & \text{if } V_{0i,j} \geq 0 \\ (V_0|\phi|)^n_{i,j} = V_{0i,j}[max(D^{+x}_{i,j}, 0)^2 + min(D^{-x}_{i,j}, 0)^2 + \\ \qquad max(D^{+y}_{i,j}, 0)^2 + min(D^{-y}_{i,j})^2]^{1/2} & \text{otherwise} \end{cases} \tag{1.46}$$

where $D^{+x}_{i,j} = (\phi^n_{i+1,j} - \phi^n_{i,j})/\Delta x, D^{+y}_{i,j} = (\phi^n_{i,j+1} - \phi^n_{i,j})/\Delta y$ and $D^{-x}_{i,j} = (\phi^n_{i,j} - \phi^n_{i-1,j})/\Delta x, D^{-y}_{i,j} = (\phi^n_{i,j} - \phi^n_{i,j-1})/\Delta y$ are the forward and backward differences respectively.

The external forces left in (1.17) contribute the third underlying static velocity field for snake evolution. Their direction and strength is based on spatial

position, but not on the snake. This motion can be numerically approximated as follows. Let $\vec{U}(x, y, t)$ denote the underlying static velocity field according to $\beta\tilde{R} - \nabla g(.)$. We check the sign of each component of $\vec{U}$ and construct one-sided upwind differences in the appropriate (upwind) direction [16]:

$$(\vec{U} \cdot \nabla\phi)^n_{i,j} = max(u^n_{i,j}, 0)D^{-x}_{i,j} + min(u^n_{i,j}, 0)D^{+x}_{i,j} + max(v^n_{i,j}, 0)D^{-y}_{i,j} + min(v^n_{i,j}, 0)D^{+y}_{i,j} \quad (1.47)$$

where $\vec{U} = (u, v)$. Thus, (1.17) is numerically solved using the schemes described above.

Bibliography

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

[2] Malladi, R., Sethian, J., and Vemuri, B., Evolutionary fronts for topology independent shape modeling and recovery, In: Proceedings of the 3rd European Conference on Computer Vision, pp. 3–13, 1994.

[3] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., Gradient Flows and Geometric Active Contour Models, In: Proceedings of the 5th IEEE International Conference on Computer Vision, pp. 810–815, 1995.

[4] Caselles, V., Kimmel, R., and Sapiro, G., Geodesic Active Contour, International Journal of Computer Vision, Vol. 22, no. 1, pp. 61–79, 1997.

[5] Malladi, R., Sethian, J., and Vemuri, B., Shape modeling with front propagation: A level set approach, IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 17, no. 2, pp. 158–175, 1995.

[6] Sapiro, G., Color Snakes, Computer Vision and Image Understanding, Vol. 68, no. 2, pp. 247–253, 1997.

[7] Xu, C. and Prince, J., Snakes, Shapes, and Gradient Vector Flow, IEEE Transactions on Image Processing, Vol. 7, no. 3, pp. 359–369, 1998.

[8] Xu, C. and Prince, J., Gradient Vector Flow: A New External Force for Snakes, In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, pp. 66–71, 1997.

[9] Xu, C. and Prince, J., Generalized Gradient Vector Flow External Forces for Active Contours, Signal Processing, Vol. 71, no. 2, pp. 131–139, 1998.

[10] Xu, C., Yezzi, J., and Prince, J., On the Relationship between Parametric and Geometric Active Contours, In: Proceedings of the 34th Asilomar Conference on Signal, Systems, and Computers, pp. 483–489, 2000.

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

[12] Comaniciu, D. and Meer, P., Mean Shift Analysis and Applications, In: Proceedings of the 7th IEEE International Conference on Computer Vision, pp. 1197–1203, 1999.

[13] Comaniciu, D. and Meer, P., Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, no. 5, pp. 603–619, 2002.

[14] Siddiqi, K., Lauziere, Y., Tannenbaum, A., and Zucker, S., Area and length minimizing flows for shape segmentation, IEEE Transactions on Image Processing, Vol. 7, no. 3, pp. 433–443, 1998.

[15] Osher, S. and Sethian, J., Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics, Vol. 79, pp. 12–49, 1988.

[16] Sethian, J., Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, 1996.

[17] Sethian, J., Curvature and the Evolution of Fronts, Communications in Mathematical Physics, Vol. 101, pp. 487–499, 1985.

[18] Osher, S. and Fedkiw, R., Level Sets and Dynamic Implicit Surfaces, Springer-Verlag, New York, 2002.

[19] di Zenzo, S., A Note on the Gradient of a Multi-Image, Computer Vision, Graphics, and Image Processing, Vol. 33, no. 1, pp. 116–125, 1986.

[20] Comaniciu, D. and Meer, P., Robust Analysis of Feature Spaces: Color Image Segmentation, In: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, pp. 750–755, 1997.

[21] Cootes, T., Taylor, C., Cooper, D., and Graham, J., Active Shape Models - their training and application, Computer Vision and Image Understanding, Vol. 61, no. 1, pp. 38–59, 1995.

[22] Osareh, A., Mirmehdi, M., Thomas, B., and Markham, R., Colour Morphology and Snakes for Optic Disc Localisation, In: Proceedings of the 6th Conference on Medical Image Understanding and Analysis, pp. 21–24, 2002.

[23] Blake, A. and Isard, M., Active Contours, Springer, London, 1998.

[24] Kass, M., Witkin, A., and Terzopoulos, D., Snakes: Active contour models, International Journal of Computer Vision, Vol. 1, pp. 321–331, 1988.

[25] Ronfard, R., Region-Based Strategies for Active Contour Models, International Journal of Computer Vision, Vol. 13, no. 2, pp. 229–251, 1994.

[26] Chakraborty, A., Staib, L., and Duncan, J., Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information, IEEE Transactions on Medical Imaging, Vol. 15, no. 6, pp. 859–870, 1996.

[27] Zhu, S. and Yuille, A., Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation, IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 18, no. 9, pp. 884–900, 1996.

[28] Paragios, N. and Deriche, R., Coupled Geodesic Active Regions for Image Segmentation: a level set approach, In: Proceedings of the 6th European Conference on Computer Vision, pp. 224–240, 2000.

[29] Paragios, N. and Deriche, R., Geodesic Active Regions: A New Framework to Deal with Frame Partition Problems in Computer Vision, Journal of Visual Communication and Image Representation, Vol. 13, no. 1-2, pp. 249–268, 2002.

[30] Yezzi, A., Tsai, A., and Willsky, A., A Fully Global Approach to Image Segmentation via Coupled Curve Evolution Equations, Journal of Visual Communication and Image Representation, Vol. 13, no. 1-2, pp. 195–216, 2002.

[31] Chop, D., Computing Minimal Surfaces via Level Set Curvature-Flow, Journal of Computational Physics, Vol. 106, pp. 77–91, 1993.

[32] Adalsterinsson, D. and Sethian, J., A Fast Level Set Method for Propagating Interfaces, Journal of Computational Physics, Vol. 118, pp. 269–277, 1995.

[33] Sethian, J., Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces, Acta Numerica, Vol. 5, pp. 309–395, 1996.

[34] Sethian, J., A Fast Marching Level Set Method for Monotonically Advancing Fronts, In: Proceedings of the National Academy of Sciences, Vol. 93, pp. 1591–1694, 1996.

[35] Paragios, N. and Deriche, R., Geodesic Active Contour and Level Set for the Detection and Tracking of Moving Objects, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, no. 3, pp. 266–280, 2000.

[36] Milne, R., An Adaptive Level-Set Method, Ph.D. thesis, Department of Mathematics, University of California, Berkeley, 1995.

[37] Weickert, J., ter Harr Romeny, B. M., and Viergener, M., Efficient and Reliable Scheme for Non-Linear Diffusion and Filtering, IEEE Transactions on Image Processing, Vol. 7, pp. 398–410, 1998.

[38] Goldenberg, R., Kimmel, R., Rivlin, E., and Rudzsky, M., Fast Geodesic Active Contours, IEEE Transactions on Image Processing, Vol. 10, no. 10, pp. 1467–1475, 2001.

[39] Fukunaga, K. and Hostetler, L., The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition, IEEE Transactions on Information Theory, Vol. IT-21, pp. 32–40, 1975.

[40] Cheng, Y., Mean Shift, Mode Seeking and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, no. 8, pp. 790–799, 1995.

[41] Danielsson, P., Euclidean distance mapping, Computer Graphics and Image Processing, Vol. 14, pp. 227–248, 1980.

[42] Borgefors, G., Distance Transformations in Arbitrary Dimensions, Computer Vision, Graphics, and Image Processing, Vol. 27, pp. 321–345, 1984.

[43] Eggers, H., Two fast Euclidean distance transformations in Z2 based on sufficient propagation, Computer Vision and Image Understanding, Vol. 69, no. 1, pp. 106–116, 1998.

[44] Gevers, T., Ghebreab, S., and Smeulders, A., Color Invariant Snakes, In: Proceedings of the 9th British Machine Vision Conference, pp. 659–670, 1998.

[45] Osareh, A., Mirmehdi, M., Thomas, B., and Markham, R., Identification of Exudate pathologies and the Optic Disc in Colour Retinal Images, British Journal of Ophthalmology, Vol. 87, pp. 1220–1223, 2003.

Questions

1. *What are the advantages of geometric snakes over their parametric counterparts?*
2. *Which are some of the key papers on the geometric snake?*
3. *How do I diffuse the region segmentation map?*
4. *Describe how weighting functions $p(.)$ and $q(.)$ behave in vector diffusion?*
5. *What parameters are there in RAGS?*
6. *How do I choose the parameter values?*
7. *What are some of the disadvantages of RAGS?*
8. *What is a good source of information for learning about level sets?*
9. *How are level sets initialised?*
10. *Is the geometric snake computationally efficient? Are there any ways to speed up the convergence?*
11. *How do I find out more about the GVF and GGVF snakes?*
12. *Describe the mean shift process.*
13. *How do I find out more about the Mean Shift segmentation method?*
14. *Who else has applied di Zenzo's method of vector gradients in the active contour literature?*
15. *How do I find out more about the optic disk application from the Results section?*

Answers

1. In parametric methods, the contour needs to be explicitly parameterised, so it is not intrinsic. Different parameterisation methods will result in slightly different shapes. More importantly, when topological problems arise, parametric methods can not handle them. Geometric snakes do not explicitly track the evolving contour and instead have built in implicit functions, hence they can split or merge freely. Contours are represented as a level set of a 3D surface. Instead of evolving the contours, the 3D surface is evolved (as shown in Figure 1.13). The contours of interest always lay in the same level set, so they can be easily extracted whenever needed. Parameterisation of the contours is only needed (if desired for further statistical shape analysis) when they are steady.

   Furthermore, geometric snakes can have larger capture areas than parametric ones. The capture range can be enlarged for parametric snakes, for example using the vector diffusion technique. However, important information may be lost in this kind of process (see Figure 1.11).

2. See references [1, 2, 3, 4, 14, 6, 9, 16]. This is not an exhaustive list, just a good starting point.

3. The region segmentation gives the segmentation map S. Vector gradient computation on S results in the region map R. Simple gradient computation on R gives ∇R, i.e, region constraints immediate to the region boundaries. In order to enlarge the region force capture area, vector diffusion is applied by solving (1.10) to diffuse the region force vectors in ∇R. The diffusion is iteratively achieved by computing the equilibrium state of (1.12). Theoretically, the diffusion can be performed in any orthogonal directions within the image plane. However, practically, x and y directions corresponding to image plane coordinates are chosen. Thus, vector components $\frac{\delta R}{\delta x}$ and $\frac{\delta R}{\delta y}$ are diffused in the image domain.

4. The functions $p(.)$ and $q(.)$ are monotonically decreasing and increasing weighting functions. Their relationship is illustrated in Figure 1.27. Function $p(.)$ effectively controls the diffusion. The larger $p(.)$ is, the more is the diffusion on the region. Function $q(.)$ preserves as much gradient information as possible on the original region along the region boundaries.

5. According to (1.17), α and β are the most important parameters in RAGS. Moreover, constant K needs to be specified when diffusing the region force. There are also parameters that are commonly needed in geometric snakes, such as the time step, level set re-initialisation step, and the snake stopping criteria.

6. The parameters α and β represent a trade-off between the constant flow force and the region force. α determines the strength of the constant

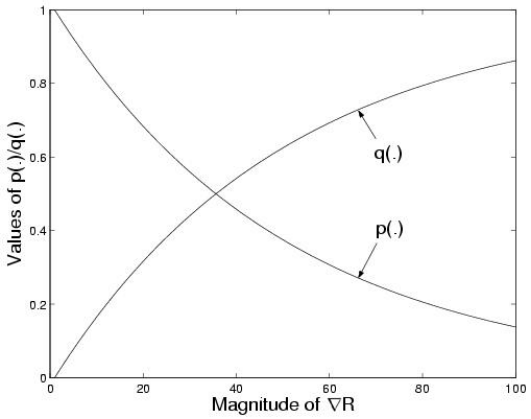

Figure 1.27: Plot of the weighting functions $p(.)$ and $q(.)$: The magnitude of ∇R is normalised to $[0, 100]$; The constant K in (1.11) is set to 50.

flow force, and also controls the inflation direction inward or outward depending on its sign. β is bounded between 0 and 1 and signifies the contribution of the region force to the formulation. When dealing with images with lots of noise and/or fuzzy boundaries, RAGS relies more on the region force. Given reasonable segmentation results, β can be chosen closer to 1. While performing region force diffusion, parameter K controls the balance between data diffusion and data conservation. The larger the value of K, the more diffusion there will be. In RAGS, K is usually selected as 50, when the magnitude of ∇R is normalised in the range of $[0, 100]$ (see Figure 1.27).

7. RAGS has to be initialised inside or outside the object of interest, just like other traditional geometric snakes. This user initialisation requirement can be a disadvantage when total automatic segmentation is needed in some applications. Also, while RAGS can perform well in the presence of noise, it does not perform so well on highly textured images, as it is designed for object segmentation based on colour. Gradient flow forces, i.e. local information, will become too noisy when computed over textured regions.

8. Sethian's book [16] and Osher's book [18] are comprehensive works on level set theory and its numerical solutions.

9. The initial level set is usually built by the signed Euclidean distance transform. However, a brute Euclidean distance transform leads to an $O(N^4)$ complexity for a $N \times N$ image, which is computationally expensive. Many algorithms have been proposed, providing approximate solutions in $O(N^2)$ time or exact solutions in $O(N^3)$ time. Interested readers are referred to [41, 42, 43].

10. Level sets trade a 2D problem into 3D. They do need more computation, especially when dealing with large images, in updating the level sets surrounding the zero level set in which the contour is embedded. There are several methods available to speed up the convergence of the active contour, such as the narrow band method [31, 32], the fast marching method [33, 34], and the AOS scheme [38].

11. See references [7] and [9].

12. The mean shift procedure can be described as follows:

    - select the window radius h,
    - select the initial seeds,
    - compute the mean shift vector in (1.40) and translate the search window by that amount,
    - repeat the mean shift computation for each seed and translate till convergence,
    - perturb the converged seeds by randomly shifting them a small amount, repeat the mean shift computation till convergence,
    - using K-NN, classify all the data points to the converged centres.

13. See references [20] and [13].

14. See references [6] and [44]. The work by Sapiro [6] has applied it to the geometric snake and that of Gevers et al. [44] has applied it to the parametric snake.

15. See references [22] and [45].