Contents

Chapter 1

An Applied Comparative Study on Active Contour Models in Mammographic Image Segmentation

Xianghua Xie
Department of Computer Science, University of Wales Swansea,
Faraday Tower, Singleton Park, Swansea SA2 8PP, UK.

Deformable contour models, also known as snakes, are commonly used in image processing and computer vision due to their natural handling of shape variation and independence of operation (once initialized), which make them highly appropriate to segment mass lesions in digital (or digitized) mammographic images. The extracted shape and texture information through contour based segmentation are useful in determining benignancy or malignancy. In this chapter, we thoroughly examine four edge based active contour models in segmenting mass lesions in mammogram images. Two of them are widely used, classic active contour models and the other two are most recent advances in active contouring. Experiments are carried out on both synthetic and real world images to compare their segmentation accuracy, as well as the ability in handling weak edges and difficult initializations.

1.1 Introduction

Breast cancer is one of the leading cause of cancer deaths in women and the risk of developing breast cancer in life time for women is very high, from eight to twelve percent [60]. Early detection is extremely important, since breast cancer can be successfully treated [35, 60] if diagnosed in the early stage. Mammography has been proved effective in examining abnormalities for early detection which is the key to improve breast cancer prognosis. Hence great effort has been put into digital mammogram analysis. Analyzing mammogram using computer vision has been widely reported in the literature, e.g. [23,38,41].

More often than not, the mammogram images are preprocessed using skin-line segmentation to isolate the breast region, e.g. [13,50,60]. In [60], fast marching method was used to delineate the breast region. Rule based fuzzy reasoning was used in [50] and in [13] active contour model was used to extract the skin line. Extracted regions of interests may also be further preprocessed, such as feature enhancement. Then various post processing tasks will be carried out. In [36], the authors provided a brief overview of image processing techniques applied to mammogram image analysis.

Segmenting mass lesions is a critical step in automatic or computed aided detecting abnormalities

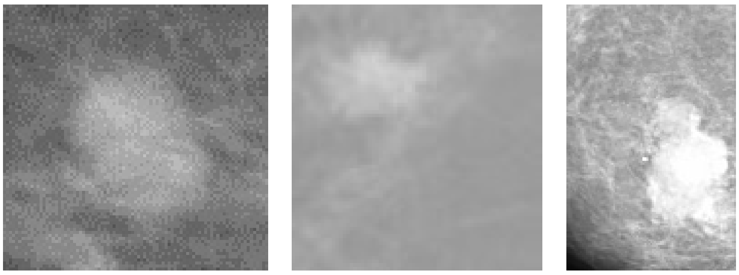

Figure 1.1: Example mass lesions in mammogram images.

and diagnosis. Masses are space occupying lesions, characterized by their shape, margin and density. A benign neoplasm is smoothly marginated, while a malignancy often has an indistinct border with low contrast which appears more spiculated over time [29]. Some example mass lesions are shown in Fig. 1.1. Potential lesion sites can be automatically or semi-automatically detected and located, e.g. [23]. These regions will be closely examined. Thus lesion segmentation is useful to delineate them from surrounding tissues. Various techniques have been developed to carry out this task. In [23], the authors used shape prior based radial gradient index and intensity profile to segment mass lesions. Simple intensity priors are used to segment lesions [5, 52]. Texture features derived from Wavelet transform, Laws filtering and Gaussian derivative filtering to detect tumor tissues in mammogram. Markov random filed has also been used to model the lesion regions [12, 17, 27, 44]. In [15], an improved region growing method which takes into account the uncertainty present around the boundary of tumor was used to segment mass lesions. Other techniques used in mammogram image segmentation include unsupervised learning using SOM [24], fuzzy sets [40], morphological process [18], watershed segmentation [41], and active contouring [38, 52].

Active contour models, also known as snakes, are commonly used in image processing and computer vision due to their natural handling of shape variation and independence of operation (once initialized), which make them highly appropriate to segment mass lesions in digital (or digitized) mammographic images. Examples works can be found in [52] where a geometric snake was used and in [38] where active contour model energy functional was minimized using genetic algorithms (GA). In this chapter, we thoroughly examine four edge based active contour models in segmenting mass lesions in mammogram images. Two of them are widely used, classic active contour models and the other two are most recent advances in active contouring. Experiments are carried out on both synthetic and real world images to compare their segmentation accuracy, as well as the ability in handling weak edges and difficult initializations.

The rest of the chapter is organized as follows. In Section 1.2, we discuss important aspects in designing an active contour model. Section 1.3 provides an overview of geometric active contour model and the level set method. In Section 1.4, we provide detailed reviews of the four edge based active contour studied in the chapter. Implementation details and numerical solutions are presented in Section 1.5. Experimental results and comparative studies are given in Section 1.6, followed by discussions on parameter settings and computational complexity in Section 1.7. Concluding remarks are given in Section 1.8.

1.2 Active Contour Design

Active contour or snake based methods have been widely used for shape recovery due to their natural handling of shape variation, e.g. [7, 33, 56]. Their design normally involves the consideration of the following three fundamental issues: contour representation and its numerical solution, object boundary description and stopping function design, and initialization and convergence.

1.2.1 Contour representation and numerical method

Contour representation involves the decision whether to use a parametric or implicit representation. Parametric active contours are usually represented as polynomials or splines, e.g. [4, 21], and interpolated using landmarks. They suffer from topological issues as it is difficult to determine the evolution of the contour crossing points and its reparameterization after every topological change, hence, predefined topology adaptations are usually necessary, e.g. [10]. The selection of landmarks is also non-trivial and critically influences shape description. Implicit representations, on the other hand, do not explicitly interpolate the contours. Instead, they embed the contour into another function, which is then temporally adapted to affect snake deformations. The Level Set method [43] is the most widely used technique to implicitly represent active contours that handle topological changes. The snake is embedded in the zero level set and its temporal evolution is achieved by deforming the entire level set function. It can naturally model contour propagation and topological changes, however, it is generally slower than parametric methods as the deformation of a higher dimensional function is necessary. Very recently Morse *et al.* [30] proposed to implicitly represent active contours using radial basis functions (RBFs) by placing RBFs at snake landmarks. This then avoids manipulating a higher dimensional function, however, it requires dynamic insertion and deletion of landmarks. Similar to parametric representation, the resolution of the landmarks can affect the accuracy of contour representation.

While choosing a representation scheme, one also needs to select appropriate numerical methods to solve the contour evolution equation. The Finite Element Method [11] is the common technique used in parametric snake models, while the Finite Difference Method [43] is mostly used in implicit snake models to numerically solve the partial differential equations (PDEs). Alternative approaches do exist, amongst them [20] which adopted the stochastic dynamics of interface propagation into the level set framework.

1.2.2 Object boundary description and stopping function design

These functions determine where the contours are expected to be attracted to and be stabilized at. Boundary description using the image gradient is the most commonly used technique e.g. [7,26,34,45,56], which assumes object separation by way of intensity discontinuities. This often results in broken edges and weak boundaries due to the lack of global information. Techniques such as anisotropic smoothing [47] can be used to enhance and refine the boundaries.

On the other hand, region-based techniques such as [9, 33, 46, 48, 61], generally use more global information to define object boundaries. Zhu and Yuille [61] proposed a region competition approach by performing statistical tests while growing multiple regions independently and using a minimum description length (MDL) criteria to minimize the overlapping of the growing regions competing for neighboring pixels. In [33], Paragios and Deriche used a bank of isotropic Gaussian filters, LoG filters, and Gabor filters to extract textural features modeled using a mixture of Gaussians. Their region-based snake segmentation then was based on the maximization of *a posteriori* probabilities. Wang and Vemuri [49] proposed an active contour model based on tensor field segmentation using a piecewise constant region model. These techniques are generally application dependent, need some prior knowledge of the objects of interests, and like other region segmentation methods are liable to over-segment or under-segment the object.

There have also been attempts to bridge boundary and region-based techniques, e.g. [8, 16, 49, 53]. For example, Chakraborty *et al.* [8] integrated region segmentation and gradient based boundary information in a Bayesian framework, while Xie and Mirmehdi [53] used diffused region forces to prevent the snake from leaking through weak edges. One of the main challenges for integrated approaches is that the region and boundary information may not locally correspond.

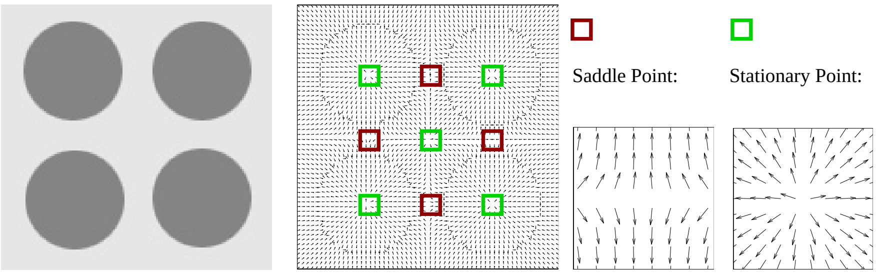

Figure 1.2: The four-disc problem - from left: An image contains four discs, the GGVF vector map, enlarged saddle point (red square) and stationary point (green square).

1.2.3 Initialisation and convergence

As snakes generally provide locally minimized solutions, it is a great challenge to achieve initialization invariancy and robust convergence, particularly for boundary based methods. The problem is compounded when the snake has to deal with complex topologies and concave shapes.

Methods that rely directly on the boundary gradient inevitably have very limited capture range and are not able to reach deeper concavities. Using the distance transform to obtain a distance vector flow (DVF) was one attempt to enlarge the capture range [11], but this still has difficulties with concave shapes. Caselles *et al.* [7] introduced the constant flow into the geometric active contour model to speed up its convergence and to pull or push the active contour towards object boundaries. However, this only monotonically expands or shrinks the contour. The weighted area functional addition to the geodesic snake by Siddiqi *et al.* [45] also did not provide a satisfactory solution to convergence on weak edges [58]. Xu and Prince [56, 57] iteratively diffused the edge gradient vector to significantly enlarge the capture range and increase the ability to reach concavities. Their snake was initialized across the object boundary due to the bidirectional nature of the vector field, where vectors point towards the object boundary from both sides. This can prevent the snake from leaking through small boundary holes or weak edges, but only to some degree. Also, bidirectionality can cause their contours to collapse on approach to the same object boundary. More importantly, Xu and Prince's GVF/GGVF active contours fail to evolve at saddle points, i.e. when the contour is tangent to the force vector. This can be demonstrated using the four-disc problem in Fig. 1.2 where there are four saddle points (indicated in dark red) and five stationary points (in light green) in the GGVF vector map (final results in Fig. 1.8). These critical points are commonly found in real images when using GVF/GGVF force fields, greatly restraining their application to real world problems.

There have been several attempts to solve the convergence issue. Paragios *et al.* [34] added the constant flow to the GVF model which pushes the active contour when it is close to tangent to the underlying vector. This however can not deal with stationary points, e.g. the center of the image in the four-disc problem in Fig. 1.2. Li *et al.* [26] suggested the active contour be split along stationary and saddle points but this strategy is inappropriate when the active contour is initialized inside or crosses the object boundary. Gil and Radeva [14] proposed the curvature vector flow (CVF), derived from a curvature dependent distance transform, to push the contour into concavities. Their method only reduces the number of stationary or saddle points and can not completely eliminate them. Moreover, *it requires there be no gaps in the object boundary.* Very recently, Li and Acton [25] proposed convolving the image edge map with a vector field kernel, which comprises radial symmetric vectors pointing towards the center of the kernel. This is equivalent to assigning an attraction force for each pixel position, based on which the snake is evolved. The authors have shown better initialization and noise insensitivity. However, it is still a stationary vector field which will suffer from the convergence issues discussed above.

Combined optimization techniques have also been used to improve initialization invariancy and convergence ability. Boykov and Kolmogorov [3] considered the segmentation problem as finding geodesics, similar to [7], but using graph cuts to find the global minimum. This exhibits less sensitivity to initialization, but graph cuts tend to choose shorter cuts which may result in inaccurate segmentation. Similar ideas have been explored by others, e.g. see [2].

Recently, Jalba *et al.* [19] proposed a charged particle model (CPM) based on electrostatics. The particles are attracted towards object boundaries due to the potential forces while being repelled by each other. The authors demonstrated that the particles could be initialized randomly across the image and did not suffer from convergence issues related to GVF/GGVF as particles once attracted to the boundaries would move along the boundary under the influence of the repulsive force. However, as shown in [59], particles on weak edges will be attracted to nearby strong edges resulting in broken contours. The method also frequently involves particle insertion and deletion, and is computationally expensive. Yang *et al.* [59] migrated this particle model to a contour model by using similar, but conditionally imposed, attraction and repellant forces. Their model showed improvements on the original CPM, the geodesic, and the GVF snakes.

In [54], Xie and Mirmehdi proposed a snake external force field based on magnetostatics. The authors hypothesized magnetic interactions between the active contours and the object boundaries, resulting in a magnetic field that can push the contours to object boundaries. A simplified snake model was applied to synthetic boundaries to illustrate the convergence ability of the model. The authors later extended the work in [55], which is reviewed in depth later in Section 1.4.

1.2.4 Some remarks

The three issues discussed above are not always independent of each other. An appropriate snake representation ensures the snake handles the deformations properly, however, representation schemes that support topological changes do not necessarily always achieve the desired contour evolutions. Better boundary description can improve convergence ability by preventing the contour from leaking through and good convergence ability is paramount to not compromise any gains from carefully chosen contour representation and boundary extraction. On the other hand, good convergence properties can compensate certain inadequacies in boundary description, e.g. bidirectional forces can stabilize contours at weak edges. In [55], the authors showed that using a novel contour evolution force can achieve significant improvement in initialization invariancy and convergence ability, even when using very simple boundary descriptions directly based on the image gradient.

1.3 Geometric Active contour and Level Set

In this section, we review the geometric active contour model and the level set method. Readers who are familiar with these topics can proceed to Section 1.4. However, this section serves as the foundation for the methods described in this chapter.

1.3.1 Geometric active contour

Geometric active contours were introduced by Caselles et al. [6] and Malladi et al. [28] 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, a discontinuity in orientation of the boundary of a shape and it can also be thought of as a zero-order continuity, from which more varied and precise representations of shapes can be derived, and a (parabolic) curvature term that smoothes the front, showing up significant features and shortening the curve.

Let $C(x,t)$ be a 2D active contour. The Euclidean curve shortening flow is given by

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

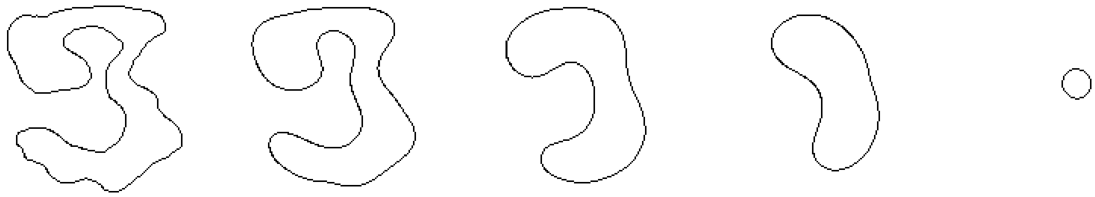

Figure 1.3: Motion under curvature flow: A simple closed curve will (become smoother and) disappear in a circular shape no matter how twisted it is.

where t denotes the time, κ is the Euclidean curvature, and $\hat{\mathcal{N}}$ 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 smoothes the evolving curve (see Figure 1.3).

In [7,22], 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 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 $g(x) \rightarrow 0$ as $x \rightarrow \infty$, and $g(x) \rightarrow 1$ as $x \rightarrow 0$. A typical function for $g(x)$ can be:

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

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

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

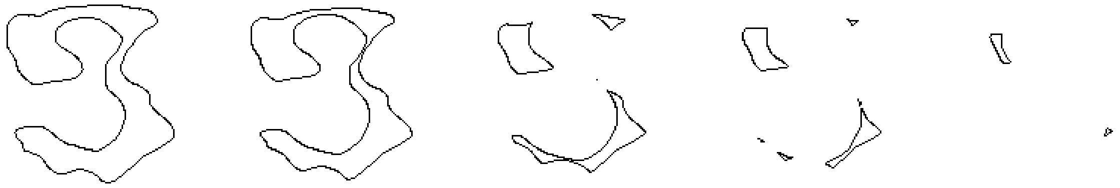

Figure 1.4: Motion under constant flow: it causes a smooth curve to evolve to a singular one.

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 modeling, 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. [7] added a constant inflation term to speed up the convergence. The constant flow is given by $C_t = \hat{\mathcal{N}}$ 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 1.4). However, integrating it into the geometric snake model lets the curvature flow (1.1) remain regular:

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

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

1.3.2 Level Set Method

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 [32,43]. 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 1.5, 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.

Let C be a level set of a function of $\phi : [0, a] \times [0, b] \to \Re$. 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 $C_t = \mathbf{F}\hat{\mathcal{N}}$ for a given function $\mathbf{F}$, then the embedding function should deform according to $\phi_t = \mathbf{F}|\nabla\phi|$, where $\mathbf{F}$ 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 the advecation force, the first term in (1.8), can be easily transferred to a level set representation:

$$\begin{cases} C_t = g(|\nabla I|)\kappa\hat{\mathcal{N}} \to \phi_t = g(|\nabla I|)\kappa|\nabla\phi| \\ C_t = g(|\nabla I|)c\hat{\mathcal{N}} \to \phi_t = g(|\nabla I|)c|\nabla\phi| \end{cases} \tag{1.9}$$

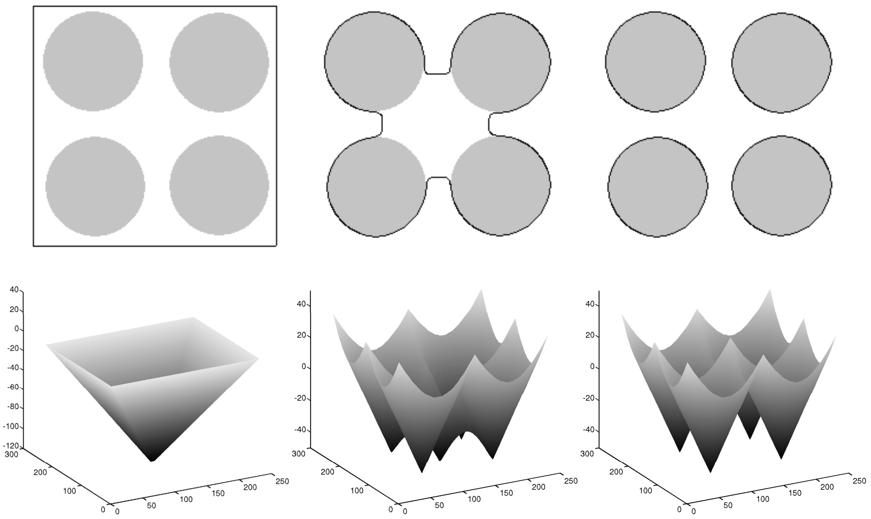

Figure 1.5: Level sets evolution for an embedded snake - top row: initial snake on test image, evolving contour, and final converged snake - bottom row: Corresponding evolving level sets. The snake is tracked at zero height.

The other external forces in (1.8) 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 $\hat{\mathcal{N}}$ is the inward normal, the level set representation of the inward unit normal is given by

$$\hat{\mathcal{N}} = -\frac{\nabla\phi}{|\nabla\phi|} \tag{1.10}$$

Then, we have

$$\mathbf{F}\cdot\hat{\mathcal{N}} = -\frac{1}{|\nabla\phi|}(\mathbf{F}\cdot\nabla\phi) \tag{1.11}$$

Combining (1.9) with (1.11), the level set representation of (1.8) is given by:

$$\phi_t = g(|\nabla I|)(\kappa + c)|\nabla\phi| + \nabla g(|\nabla I|)\cdot\nabla\phi \tag{1.12}$$

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

$$\kappa = \nabla\cdot\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) = \frac{\phi_{xx}\phi_y^2 - 2\phi_y\phi_x\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}} \tag{1.13}$$

1.4 Edge Based Active Contour Models

Here we review four edge based deformable models which we shall later apply to for comparison in mammogram image segmentation. All the models come with certain advantages and disadvantages. In what follows, f is the gradient magnitude of image I, $\mathbf{x}$ is a pixel position, $\hat{\mathcal{N}}$ is the unit inward normal of the evolving contour, and κ denotes the curvature.

1.4.1 Geodesic snake

This model has been detailed in the previous section. To briefly summarize, the image dependent constant flow can be formulated as $C_t = g(\mathbf{x})\hat{\mathcal{N}}$. This shows each point on the contour moving in the direction of its normal at a speed proportional to $g(\mathbf{x})$. It monotonically shrinks or expands the contour towards the object boundary and plays an important role in both the geodesic snake model introduced by Caselles *et al.* [7], and in later incremental improvements, such as [45]. The geodesic snake formulation is given in (1.8) and its level set representation takes this form:

$$\Phi_t = g(\mathbf{x})\nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi| + cg(\mathbf{x})|\nabla\Phi| + \nabla g(\mathbf{x}) \cdot \nabla\Phi. \tag{1.14}$$

1.4.2 Generalised gradient vector flow (GGVF) snake

The parametric GVF [57] and its generalized version GGVF [56] introduced by Xu and Prince have been widely used in active contour models as external forces, e.g. [39]. The GGVF field, $\tilde{\mathbf{v}}(\mathbf{x})$, is defined as the equilibrium solution of the partial derivatives of:

$$\mathbf{v}_t(\mathbf{x}) = p(|\nabla f(\mathbf{x})|)\Delta\mathbf{v}(\mathbf{x}) - q(|\nabla f(\mathbf{x})|)(\mathbf{v} - \nabla f(\mathbf{x})), \tag{1.15}$$

where the initial vector field $\mathbf{v}(\mathbf{x}, t = 0) = \nabla f(\mathbf{x})$, and $p(.)$ and $q(.)$ are monotonically non-increasing and non-decreasing functions respectively, controlling the amount of diffusion. These two functions are selected such that $p(.)$ gets smaller as $q(.)$ becomes larger with the desired result that in the proximity of large gradients, there will be very little smoothing and the vector field will be nearly equal to $\nabla f(\mathbf{x})$, i.e. $p(|\nabla f(\mathbf{x})|) = e^{-\frac{|\nabla f(\mathbf{x})|}{K}}$ and $q(|\nabla f(\mathbf{x})|) = 1 - p(|\nabla f(\mathbf{x})|)$, where K is a constant and acts as a trade-off between field smoothness and gradient conformity. The GGVF active contour can then be defined as:

$$C_t = \alpha g(\mathbf{x})\kappa\hat{\mathcal{N}} + (1-\alpha)(\tilde{\mathbf{v}}(\mathbf{x}) \cdot \hat{\mathcal{N}})\hat{\mathcal{N}}. \tag{1.16}$$

Its level set representation then takes this form:

$$\Phi_t = \alpha g(\mathbf{x})\nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi| - (1-\alpha)\tilde{\mathbf{v}}(\mathbf{x}) \cdot \nabla\Phi. \tag{1.17}$$

1.4.3 Geodesic GVF/GGVF (GeoGVF/GeoGGVF) snake

In [34], Paragios *et al.* integrated the GVF with the geodesic active contour model, referred to here as GeoGVF. The GVF provides the bidirectional force to drive the contours towards edges, while the constant flow from the geodesic model poses conditional inflation or deflation forces on the contour. The two forces are mutually exclusive [34]:

$$C_t = g(\mathbf{x})\left(\alpha\kappa + H(\mathbf{x}) + (1 - |H(\mathbf{x})|)\tilde{\mathbf{v}}(\mathbf{x}) \cdot \hat{\mathcal{N}}\right)\hat{\mathcal{N}}, \tag{1.18}$$

where the weighting function $H(\mathbf{x})$ is:

$$H(\mathbf{x}) = sign(\tilde{\mathbf{v}}(\mathbf{x}) \cdot \hat{\mathcal{N}})e^{-\delta|\tilde{\mathbf{v}}(\mathbf{x})\cdot\hat{\mathcal{N}}|}, \tag{1.19}$$

and δ is a scaling factor. Its level set representation takes this form:

$$\Phi_t = \alpha g(\mathbf{x})\nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi| + g(\mathbf{x})H(\mathbf{x})|\nabla\Phi| - g(\mathbf{x})(1 - |H(\mathbf{x})|)\tilde{\mathbf{v}} \cdot \nabla\Phi. \tag{1.20}$$

The additional adaptive bidirectional constant force is designed to determine the curve propagation when the GVF term becomes inactive, e.g. at saddle points. GGVF generally outperforms GVF [56], and as such we implement this method with GGVF instead of GVF. We shall refer to it as GeoGGVF.

1.4.4 MAC Model

The MAC model [55] is one of the most recent advances in edge based active contouring. It has shown significant improvements over other edge based models and comparable performances against more sophisticated region based methods. Next, we present the MAC model in detail.

As demonstrated in [14,26,56–58], bidirectionality is a very useful feature for an external force field as it facilitates more arbitrary, cross boundary initialization and improves the snake's performance towards weak edges and broken boundaries. However, the vector force fields in all these works are stationary. This means almost inevitably, there exist critical points, such as saddle points, that prevent the snake from continued propagation towards object boundaries.

Combining multiple force fields each with its different influences on a contour's evolution is a common way to improve snake evolution, e.g. [34, 59]. In both of these works, an external force field is pre-computed as the basis for manipulating snakes, i.e. the GVF in [34] and the attraction force field in [59]. These force fields are stationary and independent of the snake evolution. Later, as their snake deforms, these force fields are modified according to the snake's position. In the case of [34] a weighted balloon force, borrowed from the geodesic snake, is used to influence the contour deformation. However, the direction of the balloon force at each contour position is merely a projection of the GVF vector on to the snake normal. This combination can not sufficiently resolve the problems related to the vector field itself. In [59], an image dependent competition force exerted by the snake itself as it moves is combined with the stationary attraction force to obtain a dynamic force field. However, the balance between the attraction force and the competition force in this technique is critical for successful convergence and sometimes requires careful tuning.

In the MAC model, a novel approach is taken to define a *single* external bidirectional force field which has the ability to dynamically update itself while the contour evolves. This is significantly different from other works and it does not involve the introduction of any extra forces to mobilize the force field. In brief, it is hypothesized that electric currents flowing through both the object boundary and the active contour. The magnetic fields generated by each of the currents will interact and cause a force between them. This magnetic vector field behaves as an external force to push or pull the contour towards the object boundary. Although the magnetic field caused by the image gradient is stationary, the force it imposes on the snake is dependent on the snake's evolution. Thus *the force field is dynamically changing along with the snake.* We show that by using this electromagnetic interaction analogy the snake is then much less sensitive to its initial position and much more robust towards complex geometries and topologies. Next, we present the basic theory of magnetostatics which is used to provide the analogy for MAC. The Level Set method is then used to implicitly represent and evolve MAC.

The magnetostatic field

Consider two points P and Q with charges, q_P and q_Q, and velocity vectors $\mathbf{u}_P$ and $\mathbf{u}_Q$ respectively. The magnetostatic force exerted upon q_P due to q_Q is [51]:

$$\begin{aligned} \mathbf{F}_{QP} &= q_P \mathbf{u}_P \times \mathbf{B}_P, \\ \mathbf{B}_P &= \mu_0 q_Q \mathbf{u}_Q \times \frac{\hat{\mathbf{R}}_{QP}}{4\pi R_{QP}^2}, \end{aligned} \tag{1.21}$$

where $\mathbf{B}_P$ is known as the magnetic flux density at point P due to the point charge at Q, μ_0 is the permeability constant, R_{QP} is the distance between the two charges, $\hat{\mathbf{R}}_{QP}$ is their unit distance vector, and × denotes the cross product. The magnetic field $\mathbf{H}$ is then defined as $\mathbf{H} = \mu_0^{-1}\mathbf{B}_P$. It is clear that $q\mathbf{u}$ can be viewed as an elementary current introduced by a moving charge. In order to obtain the magnetic field introduced by a current, we consider infinitesimal current segments $d\mathbf{l}_P$ and $d\mathbf{l}_Q$ at P and Q on loops C_1 and C_2 with electric currents $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively (see Fig. 1.6). The total force on $d\mathbf{l}_P$ due to current $\mathcal{I}_2$ is then:

$$d\mathbf{F}_P = \mathcal{I}_1 d\mathbf{l}_P \times \mathbf{B}_P, \tag{1.22}$$

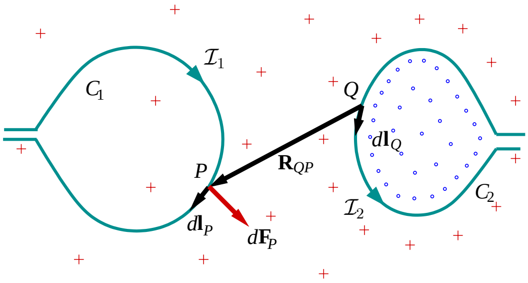

Figure 1.6: Magnetic force between two currents - see text for details.

where $\mathbf{B}_P$ is re-written as:

$$\mathbf{B}_P \equiv \frac{\mu_0}{4\pi}\mathcal{I}_2 \oint_{C_2} d\mathbf{l}_Q \times \frac{\hat{\mathbf{R}}_{QP}}{R_{QP}^2}, \tag{1.23}$$

commonly known as the Biôt-Savart Law. It defines how the magnetic flux density is obtained from a given current. In Fig. 1.6, the red '+' symbols denote the magnetic field going perpendicularly into the image plane and the blue '·' symbols denote the magnetic field coming out of the image plane. These symbols represent the overall magnetic field due to current $\mathcal{I}_2$. The magnetic force $d\mathbf{F}_P$ at position P on C_1 is in its outward normal direction. Next, we make an analogy of these principles in our active contour model.

The Basic MAC model

Instead of assigning fixed charges as in the CPM model [19], we allow the charges flow through the edges. These flows of charges will then generate a magnetic field. The active contour, carrying similar flow of charges, will be attracted towards the edges under this magnetic influence. Magnetostatic theory described earlier is now applied to our active contour model by charging both the object boundary and the active contour with electric currents. The concept is represented again in Fig. 1.6 by treating C_1 as an active contour and C_2 as an object boundary. As we are interested in the deformation of the contours, only the forces exerted by the object boundary on the active contour are needed. In other words, we shall ignore magnetic forces introduced by the active contour. We then obtain an estimation of the hypothesized direction of the currents in the object boundary and the active contour, and compute the resulting magnetic flux density at each pixel position in the image using (1.23). Based on this field the active contour drives towards the object boundary according to (1.22).

Without losing generality, let us consider the image plane as a $2D$ plane in a $3D$ space whose origin coincides with the origin of the image coordinates. Additionally, the third dimension of this $3D$ space is considered perpendicular to the image plane.

The direction of current for the object boundary can be computed using boundary orientation estimation. Let $I_x(\mathbf{x})$ and $I_y(\mathbf{y})$ be the partial derivatives in x and y for image I. This gradient vector field, $(I_x(\mathbf{x}), I_y(\mathbf{y}))$, contains vectors that are locally perpendicular to edges. Thus, the boundary orientation, $\mathbf{O}(\mathbf{x})$, can be conveniently obtained by a 90° rotation in the image plane of the normalized gradient vectors $(\hat{I}_x(\mathbf{x}), \hat{I}_y(\mathbf{x}))$:

$$\mathbf{O}(\mathbf{x}) = (-1)^{\lambda}(-\hat{I}_y(\mathbf{x}), \hat{I}_x(\mathbf{x}), 0), \tag{1.24}$$

where $\lambda = 1$ gives an anti-clockwise rotation in the image coordinates, and $\lambda = 2$ provides a clockwise rotation. This is then assigned as orientation of current in the object boundary.

In a similar fashion, the direction of current, denoted as $\boldsymbol{\Upsilon}$, for the active contour is given based on the estimation of the contour orientation. However, as we are using a level set representation, the contour is already embedded in a 3D surface Φ, obtained from the signed distance transform on the contour. No extra preparation is therefore necessary other than similarly rotating the gradient vector

$\nabla\Phi$ of the level set function either in a clockwise or anti-clockwise direction. Similar to $\mathbf{O}$, $\mathbf{\Upsilon}$ is also three dimensional and lies in the image domain, i.e. $\mathbf{\Upsilon}(\mathbf{x}) = (-\hat{\Phi}_y(\mathbf{x}), \hat{\Phi}_x(\mathbf{x}), 0)$. As a result, these rotated vectors with hypothesized electric currents will interact with each other according to (1.22) and (1.23) as illustrated in Fig. 1.6. The rotated gradient vectors given in (1.24) attract nearby contour segments that have the same current direction and repulse those that have the opposite. This means a value of 1 or 2 for λ results in opposing contour propagation.

Next, we can work out the magnetic flux density $\mathbf{B}(\mathbf{x})$ at each pixel position $\mathbf{x}$ due to the electric current applied to the object boundary. Note only pixels on the object boundary will contribute to the magnetic field. Let $\mathbf{S}$ denote the set containing all the boundary pixels and $\mathbf{s}$ denote a boundary pixel. So, given current $I_{f(\mathbf{s})}$, the magnetic flux density is computed as:

$$\mathbf{B}(\mathbf{x}) = \frac{\mu_0}{4\pi} \sum_{\mathbf{s}\in\mathbf{S}, \mathbf{s}\neq\mathbf{x}} I_{f(\mathbf{s})} \mathbf{\Gamma}(\mathbf{s}) \times \frac{\hat{\mathbf{R}}_{\mathbf{xs}}}{R^2_{\mathbf{xs}}}, \tag{1.25}$$

where $\mathbf{\Gamma}(\mathbf{s})$ is the electric current vector at $\mathbf{s}$ and proportional to the edge strength, that is, $\mathbf{\Gamma}(\mathbf{s}) = f(\mathbf{s})\mathbf{O}$, $\hat{\mathbf{R}}_{\mathbf{xs}}$ is the 3D unit vector from $\mathbf{x}$ to $\mathbf{s}$, and $R_{\mathbf{xs}}$ is the distance between them. Then, given the current I_C applied to the active contour C, its force field due to the magnetic field is:

$$\mathbf{F}(\mathbf{c}) = I_C \mathbf{\Upsilon}(\mathbf{c}) \times \mathbf{B}(\mathbf{c}), \tag{1.26}$$

where $\mathbf{\Upsilon}(\mathbf{c})$ denotes the electric current vector on the active contour position $\mathbf{c}$ and is the rotated unit vector from $\nabla\Phi$. Note $\mathbf{F}(\mathbf{c})$ is always perpendicular to $\mathbf{\Upsilon}(\mathbf{c})$, i.e. the force is always imposed in the contour normal inward or outward direction. Thus, the propagating contour will not suffer from saddle and stationary point issues, e.g. it will deal successfully with the four-disc problem. As the snake is embedded in the level set function and no longer considered as an explicit function, we also need to extend the forces imposed on the snake itself across the image domain in order to deform the level set function. This can be achieved using a force extension method such as [1], however, in this study we can simply treat each level set in Φ as a snake and compute its forces using (1.26). Thus, $\mathbf{F}(\mathbf{c})$ is extended to $\mathbf{F}(\mathbf{x})$, that is, across the image.

Note from (1.25) and (1.26), we can see that $\mathbf{B}$ intersects the image plane perpendicularly and $\mathbf{F}$ is always perpendicular to both $\mathbf{\Upsilon}$ and $\mathbf{B}$. Thus, $\mathbf{F}$ also lies in the image domain and its third element equals to zero. For simplicity, from now on, we shall ignore its third dimensional component and denote $\mathbf{F}(\mathbf{x})$ as a 2D vector field in the image domain.

Given the force field $\mathbf{F}(\mathbf{x})$ derived from the magnetic interactions between the object boundary and the active contour, the evolving active contour under this force field is:

$$C_t = (\mathbf{F}(\mathbf{x}) \cdot \hat{\mathcal{N}})\hat{\mathcal{N}}. \tag{1.27}$$

As contour smoothing is usually desirable, the standard curvature flow is added to rewrite the full MAC contour evolution formulation as:

$$C_t = \alpha g(\mathbf{x})\kappa\hat{\mathcal{N}} + (1-\alpha)(\mathbf{F}(\mathbf{x}) \cdot \hat{\mathcal{N}})\hat{\mathcal{N}}, \tag{1.28}$$

where $g(\mathbf{x})$ is the stopping function as before. Its level set representation then takes this form:

$$\Phi_t = \alpha g(\mathbf{x})\nabla \cdot \left(\frac{\nabla\Phi}{|\nabla\Phi|}\right)|\nabla\Phi| - (1-\alpha)\mathbf{F}(\mathbf{x}) \cdot \nabla\Phi. \tag{1.29}$$

The external forces in GVF and GGVF, (1.16), are static with direction and magnitude based on spatial position only, hence the contour in these cases can not propagate through stationary or saddle points. In the geodesic model (see (1.8)) direction is solely based on the active contour but magnitude is based on the boundary, thus it can only monotonically expand or shrink and can not recover broken boundaries or be initialized across boundaries. The GeoGGVF (see (1.18)) is dominated by the external

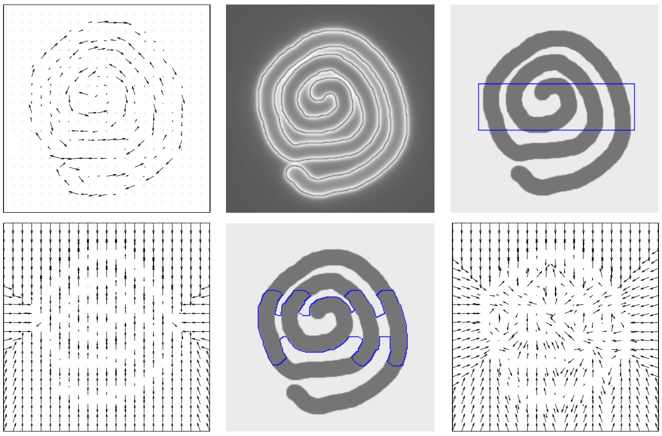

Figure 1.7: From top left: estimated boundary orientation, magnitude of the magnetic field, initial active contour, initial external force field, evolving contour, and its associated external force field. Final result is shown in the penultimate row of Fig. 1.8.

GGVF force but is conditionally influenced by a constant force imposed on the contour, however as the direction of the constant force is mainly determined by the external vector flow, it still can not resolve the topological issues of the static external force. In contrast to these models, MAC has the distinctive feature in that its direction relies on both spatial position and the evolving contour. The external bidirectional force is dynamically adaptive based on contour position (see 1.26), and equally importantly its direction is always normal to the evolving contour. The strength is determined by the magnitude of the magnetic field, thus the snake can be initialized across the object boundary and reach into concavities. Fig. 1.7 illustrates MAC's adaptive external force field.

Edge preserving magnetic force field diffusion

In keeping with other snake models based on image gradients, MAC can also suffer from image noise. Here, we perform anisotropic diffusion based on the idea in [56,57] for the GVF/GGVF snakes to refine the magnetic force field when necessary. However, we modify the diffusion functions in order to better preserve edges and show that only diffusing a scalar field, rather than a vector field, is sufficient. Thus, there is also a gain in computational efficiency.

Note that the hypothesized magnetic forces are 3-dimensional in nature but perpendicular to the image plane, and we are only interested in the forces in the image plane. So, the magnetic flux density in (1.25) only has a non-zero component which is perpendicular to the image plane and we only need to diffuse the flux density along this direction. Thus, diffusing this component, denoted as $B(\mathbf{x})$, is equivalent to diffusing $\mathbf{B}(\mathbf{x})$ in the image plane. The diffused field $\hat{\mathcal{B}}(\mathbf{x})$, is the equilibrium solution of:

$$\mathcal{B}_t(\mathbf{x}) = p(B(\mathbf{x}))\Delta\mathcal{B}(\mathbf{x}) - q(B(\mathbf{x}))(\mathcal{B}(\mathbf{x}) - B(\mathbf{x})), \tag{1.30}$$

where $\mathcal{B}(\mathbf{x})$ is initially set to $B(\mathbf{x})$, and $p(B(\mathbf{x}))$ and $q(B(\mathbf{x}))$ are given as:

$$\begin{aligned} p(B(\mathbf{x})) &= e^{-\frac{|B(\mathbf{x})|f(\mathbf{x})}{K}}, \\ q(B(\mathbf{x})) &= 1 - p(B(\mathbf{x})), \end{aligned} \tag{1.31}$$

Weighting the flux density magnitude with $f(\mathbf{x})$ in the diffusion term, $p(B(\mathbf{x}))$, ensures there is as little diffusion as possible at the object boundaries, while homogeneous areas will generally have small flux density due to a lack of support from edges, resulting in substantial diffusion.

1.5 Implementation

In this section, we provide implementation details of the MAC model, numerical solutions to GVF/GGVF and GeoGVF/GeoGGVF, and finite difference method based numerical solutions for the level set implementation that is applicable to all the four methods.

1.5.1 MAC implementation

Object boundary description may be obtained using any standard edge detector, such as the Sobel filter as used in this paper. Some erroneous edges can be removed by dropping those with magnitude less than a very small percentage, say 10% , of the maximum magnitude. This also reduces later computational costs.

The main computational costs lay in two stages: evolving the level set function and computing the magnetic flux density. The Narrow Band approach [43] is used in order to save on costs associated with updating the level set function. The summation involved in the magnetic flux density computation (see (1.25)) is generally computationally expensive. However, this can be significantly reduced by letting only edge pixels be involved in the summation. More savings can be made by approximating the flux density based on interactions in a fixed radial distance R_c, instead of across the whole image, i.e. by replacing $\mathbf{s} \in \Omega_{f(s)}$ with $\mathbf{s} \in \Omega_{R_c}$ in (1.25). R_c should be reasonably large to prevent null flux density. In this paper, we used $R_c = 100$ throughout the experiments and found it a good tradeoff between accuracy and speed.

Alternatively, a slightly less precise but an even faster approach can be used based on the relationship between the magnetic field and the magnetic vector potential. The magnetic flux density can be calculated based on its magnetic vector potential $\mathbf{A}(\mathbf{x})$ [51]:

$$\begin{aligned} &\mathbf{B}(\mathbf{x}) = \nabla \times \mathbf{A}(\mathbf{x}), \\ &\mathbf{A}(\mathbf{x}) = \tfrac{\mu_0}{4\pi} \textstyle\sum_{\mathbf{s}\in f(\mathbf{s}),\mathbf{s}\neq\mathbf{x}} I_{f(\mathbf{s})} \tfrac{\mathbf{\Gamma}(\mathbf{s})}{R_{\mathbf{xs}}}. \end{aligned} \tag{1.32}$$

The magnetic vector potential can be decomposed into two orthogonal terms corresponding to the image coordinates, (i, j), i.e. $\mathbf{A}(\mathbf{x}) = (A_i(\mathbf{x}), A_j(\mathbf{x}))$,

$$\begin{aligned} A_i(\mathbf{x}) &= \tfrac{\mu_0}{4\pi} \textstyle\sum_{\mathbf{s}\in f(\mathbf{s}),\mathbf{s}\neq\mathbf{x}} I_{f(\mathbf{s})} \tfrac{\gamma_i(\mathbf{s})}{R_{\mathbf{xs}}}, \\ A_j(\mathbf{x}) &= \tfrac{\mu_0}{4\pi} \textstyle\sum_{\mathbf{s}\in f(\mathbf{s}),\mathbf{s}\neq\mathbf{x}} I_{f(\mathbf{s})} \tfrac{\gamma_j(\mathbf{s})}{R_{\mathbf{xs}}}, \end{aligned} \tag{1.33}$$

where $\gamma_i(\mathbf{s})$ and $\gamma_j(\mathbf{x})$ are the two corresponding orthogonal components of $\mathbf{\Gamma}(\mathbf{s})$. Both of them take the same form as the Coulomb force potential [37]. Thus, the P^{3}M method [37] can be used to efficiently approximate the magnetic vector potential by computing short range interactions exactly and long range interactions approximately as used in [19, 59]. We also note that equations in (1.33) can each be viewed as a convolution with an inverse distance kernel. This can then be very efficiently solved in the frequency domain using FFT.

1.5.2 Numerical Solutions

Numerical solutions for GVF/GGVF

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 can be achieved by computing the equilibrium state of (1.15). The initial state of the force vector field $\mathbf{v}$ is given by the gradient of the edge map f. Simple central differences can be used to approximate ∇f, 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

$$u_t = \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} \tag{1.34}$$

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

$$\nabla^2 u = \frac{u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1} - 4u_{i,j}}{\Delta x \Delta y} \tag{1.35}$$

The solutions to partial derivatives of $v(x, y, t)$ are similar to those of $u(x, y, t)$. Thus, we have the following iterative solutions:

$$\begin{cases} u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t \Lambda \\ v_{i,j}^{n+1} = v_{i,j}^{n} + \Delta t \Omega \end{cases} \tag{1.36}$$

where,

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

and

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

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

$$\begin{cases} f_{x,ij} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} \\ f_{y,ij} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta y} \end{cases} \tag{1.37}$$

The convergence is guaranteed with the time step restriction of $\Delta t \leq \frac{\Delta x \Delta y}{4p_{max}}$. This numerical scheme can be similarly applied to magnetic force diffusion in the MAC model.

Numerical solution for the level set implementation

The initial level sets $\phi(x, t) = 0$ is 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.

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 $\phi_{i,j}^{n}$, the time derivative ϕ_t at (i, j, t_n) is approximated by forward difference as $\phi_t(i, j, t_n) = (\phi_{i,j}^{n+1} - \phi_{i,j}^{n})/\Delta t$, where Δt is a small time interval. Although the four models shown in (1.14), (1.17), (1.20), and (1.29) take in different forms, their underlying forces can be categorized 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 1.3. 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. 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

$$(g(.)\kappa|\nabla\phi|)_{i,j}^{n} = g(.)_{i,j}K_{i,j}^{n}({D_{i,j}^{0x}}^2 + {D_{i,j}^{0y}}^2)^{1/2} \tag{1.38}$$

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 (1.13):

$$(\phi_x)^n_{i,j} = D^{0x}_{i,j}, (\phi_y)^n_{i,j} = D^{0y}_{i,j} \tag{1.39}$$

$$(\phi_{xx})^n_{i,j} = \frac{\phi^n_{i+1,j} - 2\phi^n_{i,j} + \phi^n_{i-1,j}}{\Delta x^2}, (\phi_{yy})^n_{i,j} = \frac{\phi^n_{i,j+1} - 2\phi^n_{i,j} + \phi^n_{i,j-1}}{\Delta y^2} \tag{1.40}$$

$$(\phi_{xy})^n_{i,j} = \frac{\phi^n_{i+1,j+1} - \phi^n_{i-1,j+1} - \phi^n_{i+1,j-1} + \phi^n_{i-1,j-1}}{4\Delta x\Delta y} \tag{1.41}$$

2. The second is expanding or shrinking with a spatially constant speed, characterized by $cg(.)$ 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. It must be approximated through *entropy-satisfying schemes* [43]. Let V_0 be the constant speed function regarding to $cg(.)$. Following Sethian's *upwinding* finite difference scheme, the solution is given by

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

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.

3. The third type of motion ($\nabla g(.)$ for geodesic, $\tilde{\mathbf{v}}(.)$ for GVF/GGVF and GeoGVF/GeoGGVF, and $\mathbf{F}$ for MAC) 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.

   Let $\vec{U}(x, y, t)$ denote the underlying static velocity field. We check the sign of each component of $\vec{U}$ and construct one-sided upwind differences in the appropriate (upwind) direction [43]:

$$\begin{aligned}(\vec{U} \cdot \nabla\phi)^n_{i,j} = {} & max(u^n_{i,j},0)D^{-x}_{i,j} + min(u^n_{i,j},0)D^{+x}_{i,j} + \\ & max(v^n_{i,j},0)D^{-y}_{i,j} + min(v^n_{i,j},0)D^{+y}_{i,j}\end{aligned} \tag{1.42}$$

   where $\vec{U} = (u, v)$.

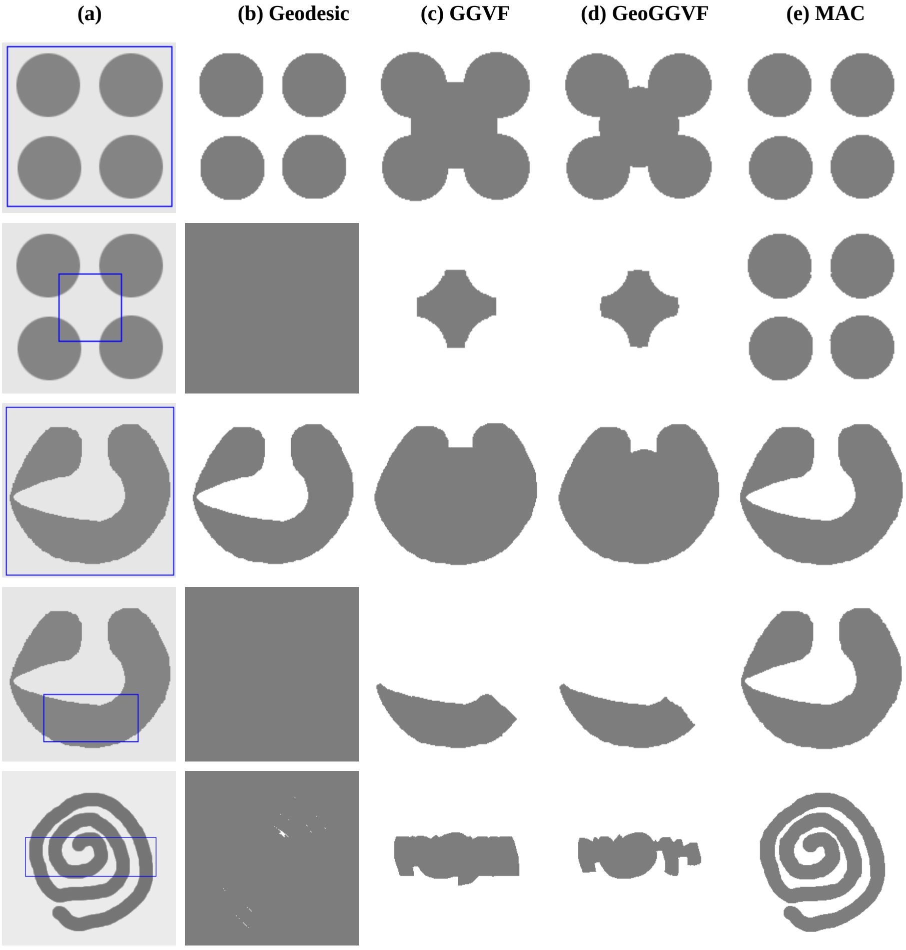

Figure 1.8: Comparing shape recovery on synthetic images (by columns) - (a) initial snakes; (b) recovered shape using geodesic, (c) GGVF, (d) GeoGGVF, and (e) MAC snakes.

By using these finite difference schemes, (1.14), (1.17), (1.20) and (1.29) can be numerically implemented. 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 [42, 43] and by Osher and Fedkiw [31].

1.6 Experimental Results

In this section, we extensively compare these four edge based active contour models. First, we closely study their convergence and initialization independence, as well as their ability to handling weak edges which frequently occur in mammogram images, using synthetic images which allows us to objectively carry out performance comparison. Next, we test the algorithms on a set of real world mammogram images to segment mass lesions. Hand labled results are used to evaluate their performance.

1.6.1 Convergence and initialization independence study

To objectively examine their initialization invariancy and convergence ability, we first present results on various shape geometries and topologies using synthetic images with well-defined object boundaries. These include the four-disc problem, acute concavities, multiple objects with complex geometry and topology, and imperfect boundaries.

The four-disc problem

Even though this is such a simple image and the active contour models are all using level set representation, convergence issues still arise. The solution becomes particularly challenging under certain initialization conditions. The first two rows in Fig. 1.8 show comparative recovered shapes for the geodesic, GGVF, GeoGGVF, and MAC models in columns (b) to (e) respectively. When the initial contour was placed outside the four discs (first row), only the geodesic snake and MAC could accurately recover them. However, in a more arbitrary cross-boundary initialization case (second row), only MAC was successful. The saddle and stationary points (see Fig. 1.2) prevented the other contours from recovering the discs. The geodesic active contour in this case simply disappeared after reaching the image borders.

Recovering acute concavities

Next, we consider the recovery of an acute concavity as shown in the third and fourth rows in Fig. 1.8, again with different initialization conditions. For the GGVF and GeoGGVF snakes, their stationary vector force fields exhibit stationary and saddle points, e.g. the saddle point at the entrance of the concave shape which prevents the snake converging to the object boundaries. Again, given an arbitrary cross-boundary initialization, the geodesic snake suffers severe problems. MAC was the only active contour model that could successfully recover the shape in both initializations.

Handling complex geometries and topologies

When dealing with complex geometries, such as the swirl shape shown in the last row in Fig. 1.8, MAC was the only model that managed to fully recover the shape.

Table 1.1 presents comparative results for the four-disc problem and recovering concavity and complex geometries with different initializations. The accuracy was measured by calculating the percentages of foreground pixels that were actually segmented as foreground (FG) and background (BG). The overall accuracy gives the measure for correctly segmented pixels given the total number of pixels in the image. The proposed MAC model significantly outperforms other boundary based snakes with a very high overall accuracy of 98.96% while the rest were well below 75%.

Handling weak edges

To further illustrate the importance of bidirectionality and dynamic force fields, Fig. 1.9 shows a circular object with a partially blurred boundary segment, commonly used for weak-edge analysis, e.g. in [53,58]. The geodesic snake in row (a) leaks through the weak edge due to the noisy stopping function. The GGVF and GeoGGVF snakes can converge to the weak edge with careful initialization due to their bidirectional force fields, however a more arbitrary initialization as seen in row (b) causes those snakes to collapse resulting in failure. MAC, on the other hand, does not suffer from these issues and can successfully localize the object, as shown in row (c).

Arbitrary initializations

In the previous examples, e.g. Fig. 1.8, we showed MAC's resilience to arbitrary initialization. Two more examples are given in Fig. 1.10. In the first row, the swirl shape is used with the initial snake just

Table 1.1: Comparative results on synthetic images (%).

		Geodesic	GGVF	GeoGGVF	MAC
Four-disc outside initialization	FG	100	100	99.2	99.3
	BG	97.9	74.0	79.5	100
	Overall	98.8	85.2	88.0	**99.7**
Four-disc cross initialization	FG	100	1.5	0.3	100
	BG	0	79.7	80.1	98.9
	Overall	43.4	45.8	45.5	**99.4**
Concavity outside initialization	FG	100	100	99.9	100
	BG	100	71.1	74.4	96.4
	Overall	**100**	83.0	84.8	97.9
Concavity cross initialization	FG	100	37.2	32.4	98.7
	BG	0	99.3	100	98.8
	Overall	41.0	75.3	74.9	**99.3**
Swirl cross initialization	FG	99.7	37.2	32.4	98.7
	BG	0	92.6	96.0	98.4
	Overall	30.3	75.8	76.7	**98.5**
Overall Average		62.70	73.03	73.98	**98.96**

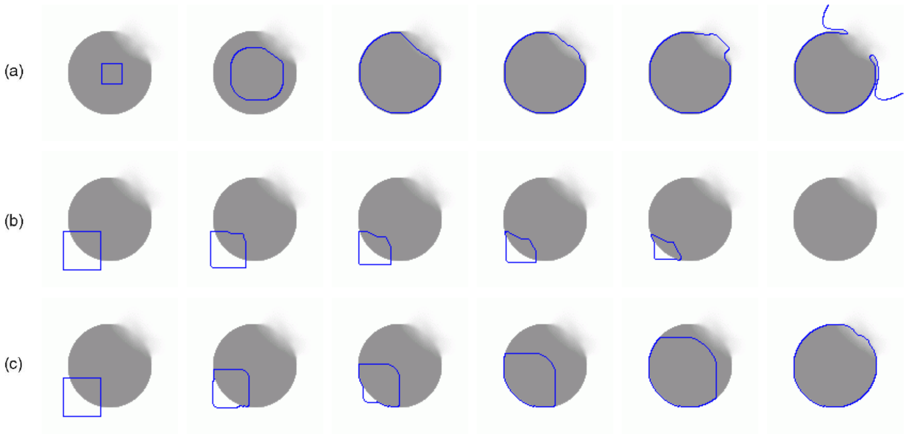

Figure 1.9: Recovering an object with weak edges - (a) geodesic, (b) GGVF, and (c) MAC snakes.

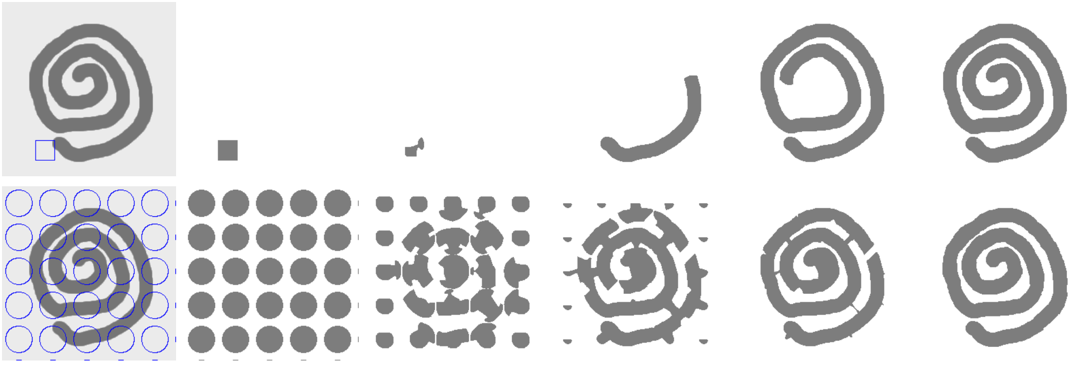

Figure 1.10: Arbitrary initializations using the MAC model.

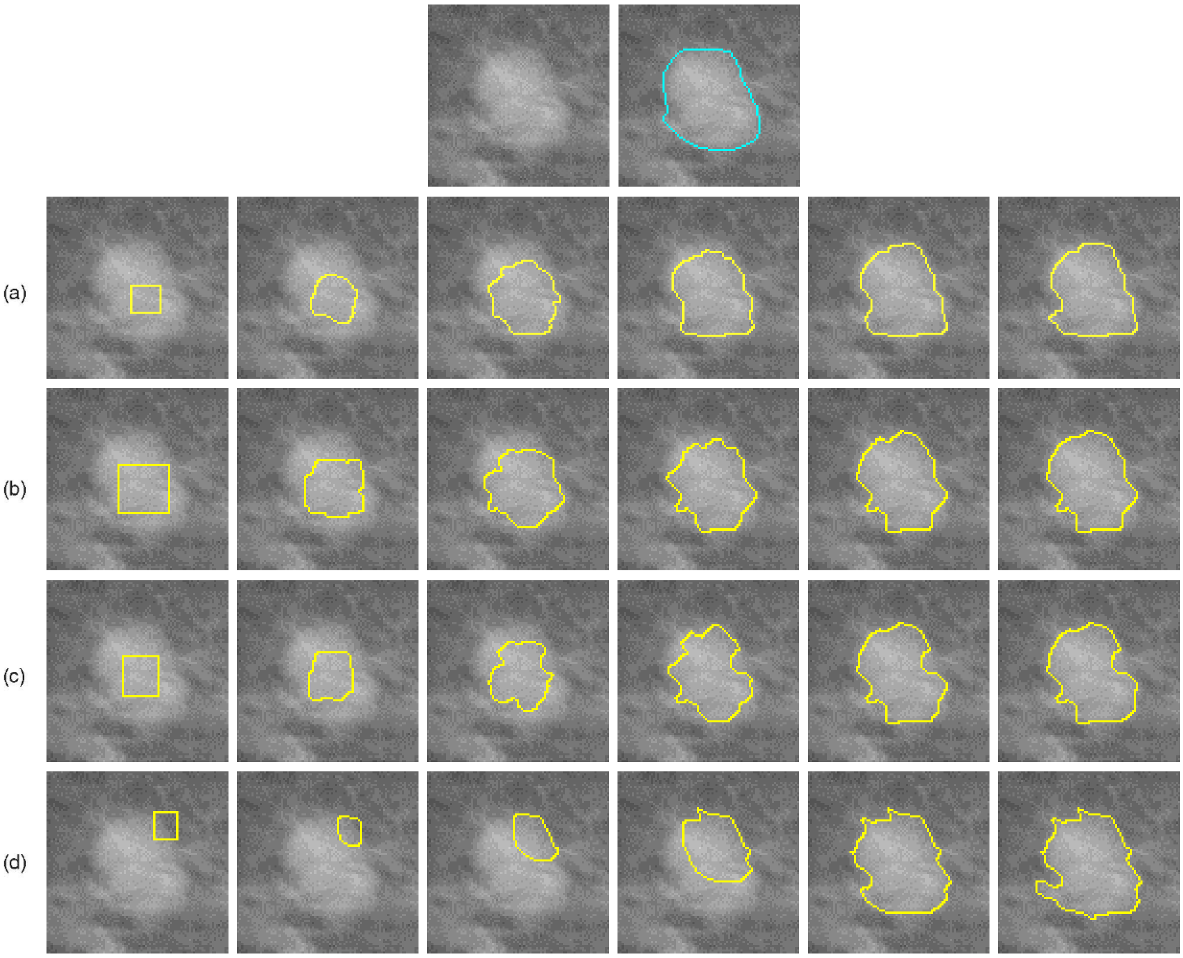

Figure 1.11: Mass lesion segmentation - Top row: mass lesion image and the hand labeled segmentation; row (a): initial geodesic snake, its evolution, and its final result; row (b): initial GGVF snake, its evolution, and its final result; row (c): initial GeoGGVF snake, its evolution, and its final result; row (d): initial MAC snake, its evolution, and its final result.

crossing the object. In the next row, several circular snakes were evenly distributed across the image domain.

These comparative analysis on synthetic images clearly show the advantages of the MAC model. Its initialization independency provides great flexibility in mammogram mass lesion segmentation since it does not require close initializations. Its ability to handle weak edges is equally important as the lesion boundaries are often blurred and indistinct.

1.6.2 Tumor and lesion segmentation

Here, we test these four edge based active contour models to segment lesions in a set of mammogram images. These lesions or potential lesions can be automatically detected using the techniques discussion earlier in Section 1.1. These mass lesion images are also hand labeled so that quantitative analysis can be carried out.

Fig. 1.11 provides the first example of mass lesion segmentation. The lesion boundaries are largely diffused, however, maintain relatively consistent contrast to the background. This allows the geodesic

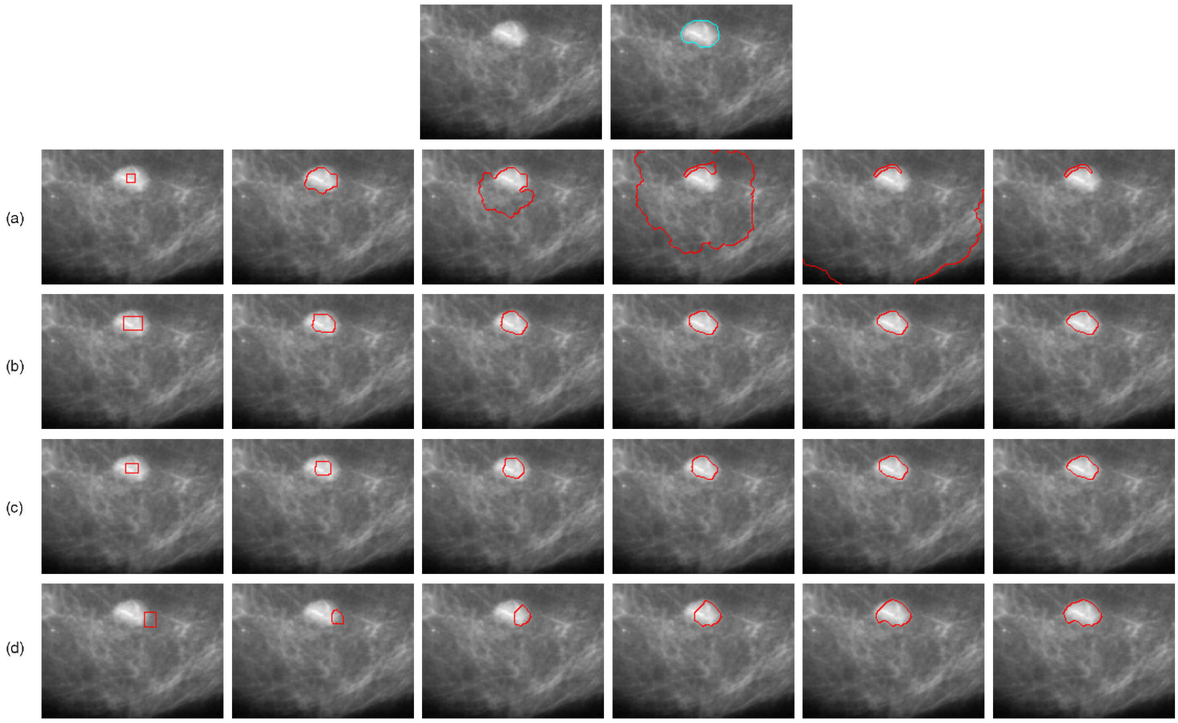

Figure 1.12: Mass lesion segmentation - Top row: mass lesion image and the hand labeled segmentation; row (a): initial geodesic snake, its evolution, and its final result; row (b): initial GGVF snake, its evolution, and its final result; row (c): initial GeoGGVF snake, its evolution, and its final result; row (d): initial MAC snake, its evolution, and its final result.

to converge to desired region boundary with careful initialization and parameter selection (mainly its constant expansion force), see row (a). A larger constant force will resulting in boundary leaking. In row (b), GGVF successfully localized the region, however careful initialization is necessary. GeoGGVF achieved similar performance as shown in row (c). The segmentation result of the MAC model was very similar to the others as shown in row (d). However, its initialization is much more challenging. The initial contour crosses the region boundary and only sits on the edge of the lesion boundary. The contour did not collapse itself but converged reasonably. None of the other techniques can achieve such result with this difficult initialization.

Another example is given in Fig. 1.12. However, this time the edge contrast around the lesion is no longer consistent and the texture is busier than the previous example. This forced the geodesic snake to choose a relatively larger constant force and as a result the contour stepped through weak edges (see row (a)). GGVF and GeoGGVF achieved good results as shown in rows (b) and (c). However, they again require dedicated initialization. The MAC model produced reasonable segmentation even with a very challenging initialization (see row (d)). This initialization independence ability of the MAC model allows automatic lesion segmentation, following automatic lesion detection. Lesion detection algorithms may not be able to find the center of the lesion and often have little knowledge of the shape of the lesion region which makes it very difficult to place the initial contours in the way that is necessary for GGVF or GeoGGVF to successfully converge. MAC, on the hand, provides great flexibility and robustness towards initialization.

More examples are given in Fig. 1.13. The geodesic, GGVF and GeoGGVF had to be carefully initialized, whileas the MAC achieved even better results without dedicated initialization. The quantitative results are shown in Table 1.2. GGVF, GeoGGVF and MAC all performed reasonably well,

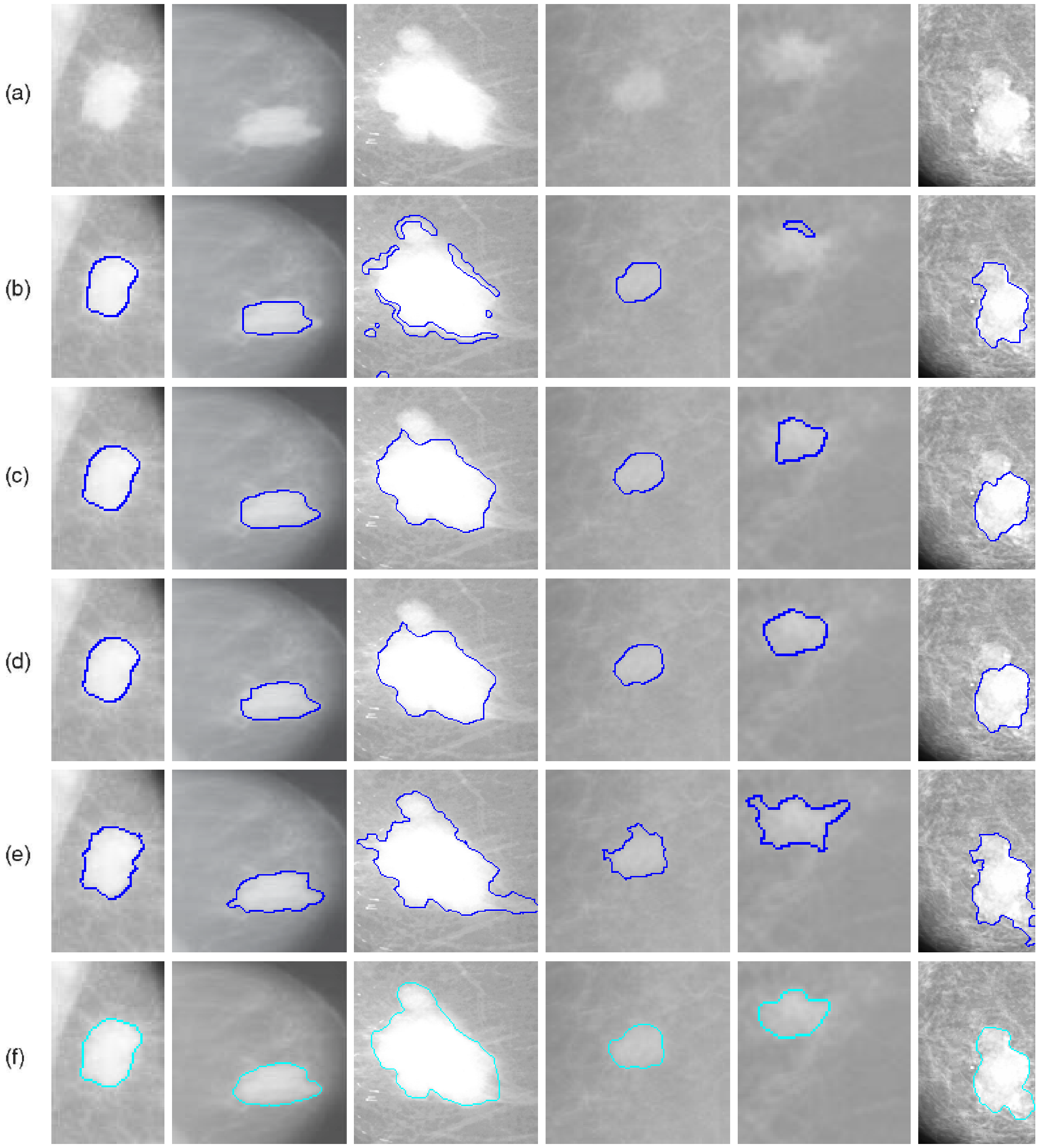

Figure 1.13: Mass lesion segmentation - Row (a): original mass lesion images row (b): geodesic results; row (c): GGVF results; row (d): GeoGGVF results; row (e): MAC results; row (f): hand labeled segmentations.

Table 1.2: Segmentation Comparison.

Test No.		1	2	3	4	5	6	7	8	**Overall**
Geodesic	spec.	100	99.98	4.04	99.96	0	99.96	0.02	99.63	62.95
	sens.	76.40	70.75	91.25	79.65	93.14	75.72	94.38	81.25	82.82
	accu.	96.34	97.35	28.52	95.02	3.14	98.56	7.68	96.23	65.36
GGVF	spec.	100	99.98	99.19	99.84	100	99.97	99.89	99.38	99.78
	sens.	88.05	85.14	82.58	80.14	64.38	83.85	69.44	65.89	77.43
	accu.	98.14	98.65	94.53	95.04	98.80	99.04	97.42	93.19	96.85
GeoGGVF	spec.	100	100	99.26	99.96	100	99.98	99.59	99.15	99.74
	sens.	87.17	81.24	81.97	78.51	62.93	82.60	86.80	71.38	79.08
	accu.	98.01	98.31	**98.41**	94.74	98.75	**98.98**	**98.55**	94.02	97.47
MAC	spec.	98.59	99.26	95.03	98.01	99.94	98.92	96.37	96.40	97.82
	sens.	97.78	95.17	96.73	92.89	89.31	97.50	98.78	98.22	95.80
	accu.	**98.47**	**98.89**	95.51	**96.76**	**99.58**	98.84	96.56	**96.74**	**97.67**

except the geodesic snake. Overall, MAC outperformed the rest.

On the application to mass lesion segmentation, we again found that MAC's ability to handle difficult initialization provided superior performance compared to the other edge based snakes. The MAC model showed great potential in automatic lesion detection and segmentation. The shape information and other feature extracted from the segmentation can be passed on for further automatic analysis. This means the whole lesion analysis process can be automated without human intervention, which can improve throughput and may reduce the possibility of false negative since more data can be processed.

1.7 Parameter settings and computational complexity

The weight for the constant force in the geodesic snake model is critical. It has to be carefully tunned to avoid weak edge leakage. The diffusion parameters for GGVF and GeoGGVF controls the degree of smoothness of the resulting vector field. Strong diffusion can result in relatively weaker edges be suppressed by nearby strong edges, which may not be desirable. If the diffusion is too small, there will be too many local minima which often leads to false convergence.

The parameters for the MAC model are α, λ, and K. Parameter α appears in all level set based snakes and is used to control the contribution from curvature flow for smoothing. For most applications it can remain a constant. Parameter λ determines the edge orientation estimated as clockwise or anti-clockwise, one of which results in localizing objects and the other in localizing their background. This is application dependent and can be determined by the user. Parameter K is only used when noise interference is significant. In general, $K = 0.2$ is large enough to diffuse noise corruption. Here, it is not used.

MAC was developed in Java on a 2.8 GHz Pentium 4 running Linux. The main computational cost occurs at two stages: computing the magnetic field and evolving the level set embedded snake. The latter is similar to that of the geodesic snake, but MAC converges faster when near edges. For the former, a direct magnetic field computation is relatively expensive, e.g. 107s for a 254×193 pixel image. This reduced to 44s when the magnetic interactions within a radial distance of 100 pixels only were computed. Alternatively, only 16s was required when using the $\mathrm{P}^3\mathrm{M}$ method or as little as 3s was needed for the FFT based method described in Section 1.5. More computational efficiency can be gained through code optimization and by implementing in a compiled language such as C.

A fair computational performance comparison for all the snakes mentioned in this paper is rather difficult, if not naive, to contemplate. Not all the snakes are successful on the same image and except for MAC, all the other snakes require specific parameter tuning and/or careful initialization. However, a simple comparison was made on a 256×256 image containing a circular object which all the snakes could fully localize with the same initialization. The computational times on the same platform for the different snake models were: 61s for the geodesic snake, 34s for the GGVF snake, 75s for the GeoGGVF snake, and 38s for MAC.

1.8 Conclusions

In this chapter, we reviewed four important edge based active contour model for mass lesion segmentation. In particular, we introduced the MAC model in detail. The MAC model uses a novel external force field which is based on hypothesized magnetic interactions between the object boundary and the active contour. It can attract the contour into deep concave regions and does not suffer from saddle point and stationary point problems. The analogy of magnetic interaction between the active contour and the object boundaries works well. However, these interactions do not necessarily have to obey physical laws. In other words, we may modify the interactions in favor of our problem domain without maintaining their physical meaning. The comparative study on both synthetic and real world mass lesion mammogram image data showed significant improvement in initialization invariancy and convergence capability of the MAC model compared to other advanced edges based methods.

Bibliography

[1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1998.

[2] R. Ardon and L. Cohen. Fast constrained surface extraction by minimal paths. *International Journal of Computer Vision*, 69(1):127–136, 2006.

[3] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision*, pages 26–33, 2003.

[4] P. Brigger, J. Hoeg, and M. Unser. B-spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, 2000.

[5] D. Cascio, F. Fauci, R. Magro, G. Raso, R. Bellotti, S. Tangaro, G. De Nunzio, M. Quarta, G. Forni, A. Lauria, M. Fantacci, A. Retico, G. Masala, P. Oliva, S. Bagnasco, S. Cheran, and E. Tores. Mammogram segmentation by contour searching and massive lesion classification with neural network. In *IEEE Nuclear Science Symposium*, pages 2695–2699, 2004.

[6] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66(1):1–31, 1993.

[7] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contour. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[8] A. Chakraborty, H. Staib, and J. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions on Medical Imaging*, 15(6):859–870, 1996.

[9] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[10] A. Clark and B. Thomas. Evolving image segmentations for the analysis of video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 290–295, 2001.

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

[12] M. Comer. Statistical segmentation of mammograms. In K. Doi, editor, *Digital Mammography*, pages 471–474. Elsevier, Amsterdam, the Netherlands, 1996.

[13] R. Ferrari, A. Frère, R. Rangayyan, J. Desautels, and R. Borges. Identification of the breast boundary in mammograms using active contour models. *Medical and Biological Engineering and Computing*, 42:201–208, 2004.

[14] D. Gil and P. Radeva. Curvature vector flow to assure convergent deformable models for shape modelling. In *IEEE Energy Minization Methods in Computer Vision and Pattern Recognition*, pages 357–372, 2003.

[15] D. Guliato, R. Rangayyan, and W. Carnielli. Segmentation of breast tumors in mammograms by fuzzy region growing. In *IEEE International Conference on Engineering in Medicine and Biology*, pages 1002–1005, 2005.

[16] J. Haddon and J. Boyce. Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:929–948, 1990.

[17] M. Haindl and S. Mikes. Unsupervised mammograms segmentation. In *IEEE International Conference on Pattern Recognition*, pages 1–4, 2008.

[18] M. Hejaz and Y. Ho. Automated detection of tumors in mammograms using two segments for classification. In *Pacific-Rim Conference on Multimedia*, pages 910–921, 2005.

[19] A. Jalba, M. Wilkinson, and J. Roerdink. CPM: A deformable model for shape recovery and segmentation based on charged particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1320–1335, 2004.

[20] O. Juan, R. Keriven, and G. Postelnicu. Stochastic motion and the level set method in computer vision: Stochastic active contours. *International Journal of Computer Vision*, 69(1):7–25, 2006.

[21] M. Kass, A. Witkin, and D. Terzopoulus. Snakes: Active contour model. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[22] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proceedings of the 5th IEEE International Conference on Computer Vision*, pages 810–815, 1995.

[23] M. Kupinski and M. Giger. Automated seeded lesion segmentation on digital mammograms. *IEEE Transactions on Medcial Imaging*, 17(4):510–517, 1998.

[24] K. Lee. Segmentation of mammography images using kohonen self-organizing feature maps. In *Midwest Artificial Intelligence and Cognitive Science Conference*, pages 41–46, 1997.

[25] B. Li and S. Acton. Vector field convolution for image segmentation using snakes. In *IEEE International Conference on Image Processing*, pages 1637–1640, 2006.

[26] C. Li, J. Liu, and M. Fox. Segmentation of edge preserving gradient vector flow: an approach toward automatically initializing and splitting of snakes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 162–167, 2005.

[27] H. Li, M. Kallergi, L. Clarke, and V. Jain. Markov random field for tumor detection in digital mammography. *IEEE Transactions on Medcial Imaging*, 14:565–576, 1995.

[28] R. Malladi, J. Sethian, and B. Vemuri. Evolutionary fronts for topology independent shape modeling and recovery. In *European Conference on Computer Vision*, pages 3–13, 1994.

[29] A. Mencattini, M. Salmeri, R. Lojacono, M. Frigerio, and F. Caselli. Mammographic images enhancement and denoising for breast cancer detection using dyadic wavelet processing. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1422–1430, 2008.

[30] B. Morse, W. Liu, T. Yoo, and K. Subramanian. Active contours using a constraint-based implicit representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 285–292, 2005.

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

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

[33] N. Paragios and R. Deriche. Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, 2002.

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

[35] R. Pfisterer and F. Aghdasi. Tumor detection in digitized mammograms by image texture analysis. *Optical Engineering*, 40(2):209–216, 2001.

[36] E. Pisano, M. Elodia, C. Bradley, B. Hemminger, M. Yaffe, R. Aylward, A. Maidment, R. Johnston, M. Williams, L. Niklason, E. Conant, L. Fajardo, D. Kopans, and M. Stephen. Image processing algorithms for digital mammography: A pictorial essay. *RadioGraphics*, 20:1479–1491, 2000.

[37] L. Polloc and J. Glosli. Comments on P3M, FMM, and the Ewald method for large periodic Coulombic systems. *Computer Physics Communications*, 95:93–110, 1996.

[38] G. Rad and M. Kashanian. Extraction of the breast cancer tumor in mammograms using genetic active contour. In *International Conference on Biomedical and Pharmaceutical Engineering*, pages 30–33, 2006.

[39] N. Ray and S. Acton. Motion gradient vector flow: External force for tracking rolling leukocytes with shape & size constrained active contours. *IEEE Transactions on Medical Imaging*, 23(12):1466–1478, 2004.

[40] M. Sameti and R. Ward. A fuzzy segmentation algorithm for mammogram partitioning. In K. Doi, editor, *Digital Mammography*, pages 471–474. Elsevier, Amsterdam, the Netherlands, 1996.

[41] V. Santos, H. Schiabel, C. Góes, and Benatti. A segmentation technique to detect masses in dense breast digitized mammograms. *Journal of Digital Imaging*, 15(1):210–213, 2002.

[42] J. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101:487–499, 1985.

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

[44] H. Sheshadri and A. Kandaswamy. Detection of breast cancer by mammogram image segmentation. *Journal of Cancer Research and Therapeutics*, 1(4):232–234, 2005.

[45] K. Siddiqi, Y. Lauzière, A. Tannenbaum, and S. Zucker. Area and length minimizing flows for shape segmentation. *IEEE Transactions on Image Processing*, 7(3):433–443, 1998.

[46] J. Suri. Two-dimensional fast magnetic resonance brain segmentation. *IEEE Engineering in Medcine and Biology*, 20(4):84–95, 2001.

[47] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. *International Journal of Computer Vision*, 68(1):65–82, 2006.

[48] X Wang, L. He, and W. Wee. Deformable contour model: A constrained optimization approach. *International Journal of Computer Vision*, 59(1):87–108, 2004.

[49] Z. Wang and B. Vemuri. Tensor field segmentation using region based active contour model. In *European Conference on Computer Vision*, pages 304–315, 2004.

[50] M. Wirth, D. Nikitenko, and J. Lyon. Segmentation of the breast region in mammograms using a rule-based fuzzy reasoning algorithm. *ICGST International Journal on Graphics, Vision and Image Processing*, 5(2):45–54, 2005.

[51] D. Wolf. *Essentials of Electromagnetics for Engineering.* Cambridge University Press, 2001.

[52] M. Xiao, S. Xia, and S. Wang. Geometric active contour model with color and intensity priors for medical image segmentation. In *IEEE International Conference on Engineering in Medicine and Biology*, pages 6496–6499, 2005.

[53] X. Xie and M. Mirmehdi. RAGS: Region-aided geometric snake. *IEEE Transactions on Image Processing*, 13(5):640–652, 2004.

[54] X. Xie and M. Mirmehdi. Magnetostatic field for the active contour model: A study in convergence. In *British Machine Vision Conference*, pages 127–136, 2006.

[55] X. Xie and M. Mirmehdi. MAC: Magnetostatic active contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):636–646, 2008.

[56] C. Xu and J. Prince. Generalized gradient vector flow external forces for active contours. *Signal Processing*, 71(2):131–139, 1998.

[57] C. Xu and J. Prince. Snakes, shapes, & gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[58] C. Xu, J. Yezzi, and J. Prince. On the relationship between parametric and geometric active contours. In *Proc. 34th Asilomar Conf. on Signal, Systems, and Computers*, pages 483–489, 2000.

[59] R. Yang, M. Mirmehdi, and X. Xie. A charged active contour based on electrostatics. In *Advanced Concepts in Vision System*, pages 173–184, 2006.

[60] R. Yapa and K. Harada. Breast skin-line estimation and breast segmentation in mammograms using fast-marching method. *International Journal of Biological, Biomedical and Medical Sciences*, 3(1):54–62, 2008.

[61] S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996.