External Forces for Active Contours via Multi-scale Vector Field Convolution

A Thesis

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

Clare Yang

August

2012

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

AUTHOR *signature*

The thesis has been read and approved by the examining committee:

Advisor

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

Month degree is awarded

August

Year

2012

Print Form

Acknowledgement

The completion of this thesis and research would not have been possible without the enduring encouragement and support I received from my advisor Dr. Scott Acton for the past several years, and for whom I would like to express my most profound gratitude. I would also like to thank my committee members Dr. John Hossack and Dr. Toby Berger for their guidance and generosity. Furthermore, I would like to acknowledge my colleagues at the VIVA Lab for the discussions we shared that led to progress of this research. Last but not the least, I thank my parents and best friend Frank Tseng for their unwavering love and trust.

Abstract

In this thesis a new method to generate the external force field for the active contour is presented. The new approach, termed multi-scale vector field convolution (MVFC), is based on the vector field convolution (VFC) method, and extends VFC to a multi-scale process that accommodates detailed, high-curvature features with a large force capture range. Via an automatic scale selection scheme, the proposed method constructs the external force field used by the active contour by varying the convolution kernel width according to local image structure. The result is an adaptive force generation method that demonstrates improved performance in terms of local convergence. Meanwhile, the new method preserves the low computation complexity as supported by traditional VFC and exemplifies as a more efficient force generation method than gradient vector flow (GVF). Synthetic and real experiments demonstrate the efficacy of MVFC in terms of detail preservation, capture range, noise resilience and low computational cost.

Table of Contents

Acknowledgement	I
Abstract	II
Table of Contents	III
List of Figures	Error! Bookmark not defined.
List of Tables	VIII
List of Symbols and Abbreviations.....	IX
1. Introduction	1
1.1 Problem Statement	1
1.2 Applications	3
2. Background.....	7
2.1 Parametric Active Contour Models.....	7
2.1.1 Image forces.....	8
2.1.2 Theory of parametric active contours	10
2.1.3 Numerical implementation of active contours	14
2.2 External Force Models for the Active Contour.....	18
2.2.1 Image gradient forces.....	18
2.2.2 Forces by distance transform	19
2.2.3 Balloon forces	21
2.2.4 Gradient Vector Flow model	23
2.2.5 Vector Field Convolution model	26
3. Multi-scale Vector Field Convolution.....	30
3.1 Scale-space Representation.....	30
3.2 Limitations of VFC	32
3.3 Multi-scale VFC.....	39
4. Results and Analysis.....	45
4.1 Synthetic Examples.....	46
4.1.1 Corner convergence	46
4.1.2. Multiple object segmentation.....	51
4.2 Segmentation using Weizmann [39, 40] Database	54

4.3	Computational Cost Analysis.....	58
5	Conclusions and Discussion	60
	Bibliography	61

List of Figures

Figure 1-1: A moment-0 image of an EVLA data cube where each pixel represents the sum brightness across the 3 GHz frequency span (23.6 to 26.6 GHz, 24012 frequency channels) in the Orion KL region. Plotting the brightness along the frequency channel axis for a particular pixel generates a spectrum. The spectrum is unique to each pixel indicating a spread of chemical composition and physical conditions. [8] 5

Figure 2-1: Analogy between electrostatic potential force and image potential force. (a): an electric charge exerts forces on a small test charge nearby and pulls it to the center of the electric field. (Image obtained online from [19]) (b): a line segment (image feature) “emits” a force field that attracts a nearby test charge (snake) towards the line segment. . 8

Figure 2-2: Example of image gradient forces. (a) A gradient force field is computed for “U” and it has limited force spread; (b) failed segmentation of the “U” shape. (Dashed contour: initial snake; solid contour: final snake.)..... 19

Figure 2-3: (a) Distance map of a binary “U” shape; (b) failed segmentation of the “U”. (Dashed contour: initial snake; solid contour: final snake); (c) distance potential forces computed for “U”..... 21

Figure 2-4: Segmentation with balloon forces. (a) Image gradient forces for “room”. (b) failed segmentation of “room” when inflation forces were added to the external force. (Dashed contour: initial snake; solid contour: final snake.) 23

Figure 2-5: Segmentation of a “U” shape with GVF snake. (a) External force field produced by GVF; (b) GVF snake successfully delineates the concave boundary of “U”. (Dashed contour: initial snake; solid contour: final snake.) 25

Figure 2-6: Configuration of a Vector Field Kernel (VFK). A magnitude function (2.19 or 2.20) is multiplied to a unit vector field kernel to create the VFK. 27

Figure 2-7: Segmentation of a “U” shape with VFC snake. (a) External force field produced by VFC; (b) VFC snake successfully segments “U”. (Dashed contour: initial snake; solid contour: final snake.) 28

Figure 3-1: Illustration of VFC external force calculated for point (x_0, y_0) . The “U” shape image is the edge map $f(x, y)$. Blue arrows mark a spatially-reversed VFK. The offsets of the VFK are marked in green. The red rectangle outlines an area in which force vectors of

the VFK is weighted by the image and summed to produce the external force at (x_0, y_0) , $F(x_0, y_0)$, marked by the red arrow. 34

Figure 3-2: Limitation of VFC at corners. (a): schematic of a spatially-reversed VFK centered on the leftmost angle of a “triangle” shape and the external force at that angle point (marked by red arrow). (b): VFC external forces generated for the “triangle” using the VFK shown in (a) and snake segmentation of the “triangle”. (Dotted oval contour is the initial snake and solid red contour is the final segmentation.)..... 36

Figure 3-3: Limitation of VFC involving two objects. (a and b): schematics of a spatially-reversed VFK shifted to the centers of two squares of different sizes (c): VFC external forces generated for the image containing two squares using the VFK shown in (a, b) and final snake segmentation of both squares. (Dotted circles are the initial snakes and solid contours demonstrate the final segmentations.)..... 38

Figure 3-4: Cross-sectional magnitude profiles of power-decaying (3.6) MVFC kernel set. It provides a force spreading range of 100 pixels and consists of 5 scales linearly spaced between $t_{\min} = 10$ pixels and $t_{\max} = 100$ pixels. Scale t is defined explicitly as the width of the function. 41

Figure 3-5: Cross-sectional magnitude profiles of Gaussian-decaying (3.7) MVFC kernel set. It provides a force spreading range of 100 pixels and consists of 5 scales exponentially spaced between $t_{\min} = 4$ pixels and $t_{\max} = 25$ pixels. Here scale t is equivalent to the standard deviation of the Gaussian. The actual width of the function, hence the force spreading range, is implemented as $4t$ 42

Figure 3-6: Illustration of MVFC. It consists of multiple versions of VFCs, each computed at a particular scale t_i ($i = 1, 2, 3 \dots$) and normalized by the total force magnitude of the field. For a common location (x_0, y_0) , an optimal scale is selected at (x_0, y_0) such that the selected scale corresponds to the maximum normalized VFC force among all the VFC forces computed at the same location. The MVFC force at (x_0, y_0) is then the found maximum VFC force of the optimal scale..... 43

Figure 4-1: Segmenting “Triangle” using VFC snakes computed at multiple scales ($t = [5, 100]$ pixels). *Top*: segmentation error (ASD) vs. VFC kernel scale. *Bottom*: segmentation results (dotted circles: initial snake; solid contours: final snake; blue arrows: VFC force field). 47

Figure 4-2: Segmenting “Triangle” using MVFC snake (dotted circles: initial snake; solid contours: final snake; blue arrows: MVFC force field). 48

Figure 4-3: (a) Three arbitrary locations (B, C, D) were selected from the image with increasing distances from the corner point A ($AB = 23$ pixels, $AC = 52$ pixels and $AD = 87$ pixels). Normalized external force magnitude across all scales at A, B, C and D are plotted. MVFC finds an optimal scale for each point where the normalized force magnitude is at maximum. 50

Figure 4-4: Optimal scales selected by MVFC across the full image. Colorbar shows the values of scales. 50

Figure 4-5: Segmenting “Two squares” with VFC snakes at multiple scales ($t = [5, 125]$ pixels). *Top*: segmentation errors (ASDs) computed separately for large (LSq) and small (SSq) squares and the sum of errors vs. VFC kernel scale. *Bottom*: segmentation results (dotted circles: initial snake; thin contours: snake evolutions; bold contours: final snake). 52

Figure 4-6: Segmenting “Two squares” using MVFC snake. (Dotted circles: initial snake; thin contours: snake evolutions; bold contours: final snake). 53

Figure 4-7: Segmentation results using MVFC, VFC and GVF-snake for four objects containing high-curvature features. (Dashed green contours: initial snake; solid red contours: final snake) 55

Figure 4-8: Segmentation results using MVFC, VFC and GVF-snake for the same objects in Fig. 4-7 but with additive noise. (Dashed green contours: initial snake; solid red contours: final snake) 57

Figure 4-9: Computational cost for image force generation using GVF and 3 different MVFC filter sets (n is number of scales per filter set) for an $N \times N$ image. 59

List of Tables

Table 4-1: Comparisons of the best-case VFC and MVFC in minimum segmentation error and force capture range for “Triangle”	48
Table 4-2: Comparisons of the best-case VFC and MVFC in segmentation error and force capture range for “Two squares”	54
Table 4-3: Segmentation accuracy (measured in ASD in pixels) for objects in Fig. 4.7. (Bold font indicates minimum error.)	56
Table 4-4: Segmentation accuracy (measured in ASD in pixels) for noisy objects in Fig. 4-8. (Bold font indicates minimum error.)	57

List of Symbols and Abbreviations

∇	Gradient operation
E_{int}	Internal energy of active contour
E_{ext}	External energy of active contour
F	External force field
f	Image edge map
I	Gray-scale image
s	Curve length parameter
t	Scale parameter
v	Vertex of active contour (Snaxel)
AC	Active Contour
ASD	Average Shortest Distance
GVF	Gradient Vector Flow
MVFC	Multi-scale Vector Field Convolution
VFC	Vector Field Convolution

1. Introduction

1.1 Problem Statement

Image segmentation is the process of subdividing a digital image into its constituent homogeneous regions. This definition is rather broad. As such, image segmentation may be carried out by analyzing image intensity gradients, image texture, high level features or even by recognition of the imaged objects such as faces or vehicles.

Perhaps no subject has received more study in multidimensional signal processing than image segmentation. The methods are numerous and the applications are ubiquitous, ranging from the microscopic to the galactic in scale. It can be said that image segmentation is the requisite first step in any automated image analysis paradigm. For example, one must delineate the boundaries of cells before such cells can be labeled as cancerous or benign. One must demarcate a tank from the background before an algorithm can recognize it as an enemy tank. And the boundaries of the left ventricle must be derived before measures such as cardiac output can be computed.

During the past two decades, the tool of active contours has emerged as a leading approach to image segmentation. The essential concept is simple: to allow a closed curve (on a two-dimensional image) to move across the image until it rests upon the underlying edges which mark a distinct object. This curve, like a rubber band, is forced, at the same time, to be smooth. Mathematical tools, more specifically, those of partial differential equations, allow such an active contour model to be implemented on actual digital images.

Shortcomings of current active contour models are addressed in this research. First, as the active contours require hundreds of small updates per image (at each discrete point in the active contour), computational complexity has been a major drawback. Second, as the active contours can be “distracted” by clutter and irrelevant objects, and as the forces derived from an object’s edges are limited in spatial extent, accurate initialization of the active contour is required (which, in the extreme, leads us back to manual segmentation). Finally, current active contours produce an implicit tradeoff between capturing large, smooth objects or fine, detailed objects. If one elects to capture coarse objects, the finer high curvature features are obliterated. On the other hand, if one desires fine objects, the active contour becomes more susceptible to noise.

The problem of computational expense is solved by using a convolution-based computation of the active contour forces. In contrast to diffusion-based force computation, as will be discussed in Chapter 2, the convolutional method is efficacious for high resolution images that need to be processed in real-time. The proposed solution, called Multi-scale Vector Field Convolution (MVFC), allows an extended capture range by using a tensor convolution to spread the edge forces across edgeless homogeneous regions. The main innovation of the thesis is the multi-scale approach, which utilizes a series of kernels that range from coarse to fine in scale. With the multi-scale approach, the tradeoff between coarse-scale segmentation and fine-scale segmentation is avoided. MFVC provides smooth boundaries with large capture ranges, while at the same time allowing high curvature features to survive the segmentation.

1.2 Applications

A. Application to medicine

One example application of the proposed segmentation method is that of delineating vascular myocytes in microscopy. Vascular myocytes are the major cell type found in a blood vessel. In vascular diseases, such as atherosclerosis, vascular myocytes proliferate and migrate out into the lumen of the blood vessel, blocking blood flow, which can then lead to stroke or a heart-attack. The most common therapy for atherosclerosis is the use of an intravascular balloon to expand and restore blood flow to the artery followed by the placement of a stent inside the artery to keep it open. In certain patients, the stented artery will close again over time as a result of high rates of vascular myocyte proliferation; a process termed in-stent restenosis [1]. Vascular myocytes are highly plastic and can be easily manipulated in a cell culture to transition from a growth arrested and contractile state to a highly proliferative and synthetic state [2]. Because of this, scientists have been able to create simple *in vitro* cell assays that mimic the process of in-stent restenosis by stimulating growth arresting myocytes with a growth factor(s) to induce proliferation [3], and design anti-proliferative drug therapies to combat atherosclerosis [1].

A challenge in this research has been the lack of accurate methods to measure cellular proliferation *in vitro* without having to terminate an experiment. A few widely used methods for counting cells and assessing proliferation in biology include the use of fluorescent probes to label DNA content, incorporation of radioisotopes, measurement of proteins associated with proliferation, or removal of the cells from the culture surface and manually counting them. All of these methods require terminating the experiment and

are very time consuming, some costly or could introduce manual bias to the result. The required task of rapidly, accurately and non-invasively counting cells calls for automated image analysis. In addition, to identify what stage the cell is during a proliferation cycle, e.g. normal stage, mitosis (cell division) and apoptosis (cell dying), it would require knowledge of the cell morphology through segmentation. This work was presented in [4].

B. Application to astronomy

A second application involves automated feature detection and segmentation in large datasets, for example those recorded in astronomical studies. Performance bottleneck is created when current practices in data analysis do not match well to the data production rate and the quality of state-of-the-art sensing instrument. National Radio Astronomy Observatory (NRAO)'s interferometer: Expanded Very Large Array (EVLA) will be running in full operation by 2015 with a bandwidth, spectral resolution and spatial resolution far exceeding many of its predecessors. A number of observations that demonstrate the early EVLA capabilities are being conducted at NRAO and the demonstration data is available to the public [5]. One such demonstration probes the Orion KL region, a chemically rich space whose "hot core" is the target of many astronomical/astrophysical studies [6, 7]. This demo data records emissions from ammonia, methanol and sulfur dioxide in the Orion KL hot core. The demo data contains 24012 spectral images covering a bandwidth of 3 GHz, and each image contains 96×96 pixels (future data is expected to have spatial resolutions far exceeding 96×96 pixels per channel image), shown in Fig. 1-1.

No table of figures entries found.

Figure 1-1: A moment-0 image of an EVLA data cube where each pixel represents the sum brightness across the 3 GHz frequency span (23.6 to 26.6 GHz, 24012 frequency channels) in the Orion KL region. Plotting the brightness along the frequency channel axis for a particular pixel generates a spectrum. The spectrum is unique to each pixel indicating a spread of chemical composition and physical conditions. [8]

The frequency correspondences of the spectral features can be used to match known spectrum absorption profiles of the molecules, which then could allow scientists to determine the spatial distributions of these molecules within the hot core and its physical conditions. It is impossible that a scientist can analyze all of the spectra data by sight within a reasonable time. To extract relevant data quickly from the data cube calls

for automated feature detection and segmentation algorithm. This work was presented in [8].

2. Background

In the background, the basics of parametric active contours, also called *snakes*, will be given. Particular attention will be paid to the derivation and numerical implementation of the active contours. Variations of the active contour and its force models will be explained in light of the rich literature.

The purpose of the background is to set the stage for the proposed multi-scale vector field convolution (MVFC) methods in terms of theoretical framework and drawbacks of current approaches.

2.1 Parametric Active Contour Models

Kass, Witkin and Terzopoulos introduced the snake model [9] to image segmentation and shape modeling in the late 1980s. Since then, the snake model has evolved into an active field of research in the computer vision community. A tremendous amount of research has sprung up from the Kass *et al.* model and has aimed at extending the robustness and effectiveness of the original snake to tasks of image segmentation [10-14] and object tracking [15-18]. In recent years, we have seen in particular a surge of activity in using snake to solve many intriguing and pressing problems of importance to medicine and other basic science.

A snake is an active contour model. It can be perceived visually as a curve that is moved across an image to capture a particular target. The mechanism giving snake the motion capability is key to understanding and designing the snake algorithm.

2.1.1 Image forces

Before we proceed with the snake algorithm, we will first define the forces associated with an image, and provide an intuitive introduction to the snake. To visualize the interaction between the snake and the feature in an image, we may draw an analogy from electrostatics. Consider an electric field induced by a negative charge shown in Fig. 2.1-a. The arrowed lines are electric field lines giving directions of flow of the electric field. In this case, the field lines converge at the negative source at the center. If we then place a free-moving positive test charge near this negative source, by Coulomb's Law, we can calculate the electric forces between the test charge and the source. The attractive forces will pull the test charge towards the source along a path given by the field line.

Figure 2-1: Analogy between electrostatic potential force and image potential force. (a): an electric charge exerts forces on a small test charge nearby and pulls it to the center of the electric field. (Image obtained online from [19]) (b): a line segment (image feature) “emits” a force field that attracts a nearby test charge (snake) towards the line segment.

Using this simple analogy, we may consider a feature of interest, such as an edge, in an image similar to the negative electric source. Instead of being a point source,

however, the feature can have an arbitrary shape, for example a line segment in black shown in Fig. 2.1-b. We can visualize this line segment emitting a field, by which we call the image potential field, such that the field lines are all directed towards the line segment. Now if we place a test charge in this field, we expect the charge to also follow the constituent vectors of the field until the charge reaches the line segment. Snake essentially behaves in the same way as the free-moving test charge in this case. When placed initially near the image feature, the snake will be pulled towards the feature by an attractive force that we call the image force. During the movement towards the image feature, we say that the snake is evolving or deforming. When the snake reaches the image feature and ceases movement, we say that the snake has converged to its target. From this example, we see that the image force is the reason behind snake evolution, and thus finding the right image force model is key to the success of the snake algorithm.

There are many ways to build image forces. To model the image force after naturally occurring forces such as electric force (or similarly gravitational force) is very intuitive; however, image force does not have to have a physical meaning. Image force is simply a mathematical construct. A common definition for image force, for example, is the gradient force:

$$\mathbf{F} = -\nabla(|\nabla I(x, y)|^2) \quad (2.1)$$

where the force is modeled by the gradient of the edge strength, which in turn is computed from the gradient magnitude of the gray-valued image. The “minus” sign in (2.1) ensures that the direction of the gradient force is pointed towards the image feature of interest. The evolution of force models for the active contour will be discussed in detail in Section 2.2.

2.1.2 Theory of parametric active contours

Basic theory for the active contour model and its numerical implementations are discussed in the remaining of Section 2.1. Contents are based on understanding and summaries from four main sources: [9, 14, 20, 21]. To keep variable notations consistent, vector quantities will be labeled in bold face.

A snake is a deformable curve. When represented using parametric equations, the snake model is a mapping of a curve length parameter s defined in range Ω to positions of the curve $\mathbf{v}(s)$ on a real plane [20]. The positions of the snake can further be written in terms of vector coordinates $[x(s), y(s)]$:

$$\Omega = [0, 1] \rightarrow \mathbb{R}^2$$

$$s \rightarrow \mathbf{v}(s) = [x(s), y(s)]$$

Typically Ω is normalized to values between $[0, 1]$. We may then define the energy terms associated with the snake. As will be discussed shortly, snake evolution is based on a minimization principle of the sum of two energy functionals.

First, we define the internal energy for the snake, expressed in (2.2). Let $\mathbf{v}'(s)$ and $\mathbf{v}''(s)$ be the first and second order derivatives of snake positions with respect to the curve length s respectively. The internal energy E_{int} is then the sum of squared derivatives computed over the total length of the snake.

$$E_{\text{int}}(\mathbf{v}(s)) = \frac{1}{2} \int_{\Omega} \alpha |\mathbf{v}'(s)|^2 + \beta |\mathbf{v}''(s)|^2 ds \quad (2.2)$$

E_{int} acts as a smoothness constraint function for the snake. One way to understand E_{int} is to consider the parameter s a time variable; we may then consider $\mathbf{v}'(s)$ as the

velocity of a particle moving on the curve, and $\mathbf{v}''(s)$ as the acceleration of the particle. A smoother curve entails that the particle would experience less motion, or equivalently the velocity and acceleration of the particle would be lower, which would then lead to a lower internal energy for the particle based on (2.2). Because natural objects usually have smoother shapes, it is intuitive to limit the value of the internal energy for the snake so that the snake can form shapes more truthful to the natural objects. The weights α and β in (2.2) control the relative importance of the first-order smoothness (elasticity) and the second-order smoothness (rigidity) of the snake [9].

Next, we define the external energy for the snake. Expressed in (2.3), the external energy, E_{ext} , is computed as the sum of squared gradient magnitudes of intensity values of pixels that belong on the snake. In simpler term, E_{ext} is the total edge strength found on the entire path of the snake.

$$E_{\text{ext}} = - \int_{\Omega} |\nabla I(\mathbf{v}(s))|^2 ds \quad (2.3)$$

It is helpful to note that E_{ext} is intrinsic to the image because the edges that give rise to E_{ext} are features on the image. E_{ext} is not created by the snake; hence it is a property “external” to the snake. E_{ext} is only dependent on snake positions because they define the domain over which E_{ext} is evaluated. As snake evolves, the domain changes, and too does the value for E_{ext} . The criterion to stop snake evolution is when the value of E_{ext} reaches a minimum. Because of the “minus” sign in (2.3), this criterion is equally satisfied when the total edge strength found on the path of the snake reaches a maximum, which becomes true when the snake moves right on top of the edge. It is assumed that the edge feature which we wish to segment produces the maximum change in pixel intensities

(equivalently the maximum edge strength) in the image. Thus the convergence criterion is not met until the snake actually moves to the target edge.

The total energy for the snake is the sum of the internal and the external energies defined in (2.2) and (2.3):

$$E_{snake}(\mathbf{v}(s)) = E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s)) \quad (2.4)$$

From previous discussions, we see that imposing a small E_{int} allows the snake to stay smooth, and the process of finding minimum E_{ext} guides the snake towards its target. Now when two energies are added and we are looking for a minimum of their sum, we are essentially seeking a balance of influences from both energies and aiming to achieve a best-case scenario of combined curve smoothness and target localization.

More formally, snake evolution seeks a curve $\mathbf{v}(s)$ that minimizes the total energy E_{snake} defined in (2.4). Conventional calculus is of no help here at finding the minimum total energy, because the energy terms associate with the snake are functionals. Both E_{int} and E_{ext} are functions of snake positions $\mathbf{v}(s)$, which in turn is another function of the parameter s . Thus By definition E_{int} and E_{ext} are functionals. To minimize a functional, a necessary step is to find the derivative of the functional with respect to the function on which the functional is dependent. We will need a more advanced mathematical tool known as calculus of variations to compute functional derivatives. [14].

Employing calculus of variations, the functional derivative of the total snake energy with respect to the curve configuration is given in (2.5). To find the snake configuration that minimizes the total energy, we set the left hand side of (2.5) equal to zero and solve for $\mathbf{v}(s)$.

$$\frac{dE_{snake}(\mathbf{v}(s))}{d\mathbf{v}(s)} = (\alpha \mathbf{v}'(s))' - (\beta \mathbf{v}''(s))'' - \nabla E_{ext}(\mathbf{v}(s)) = 0 \quad (2.5)$$

The above equation is known as the Euler-Lagrange Equation for the snake. At minimum of (2.4), the curve must satisfy this equation.

The Euler-Lagrange Equation can also be rewritten as the following:

$$\mathbf{F}_{int}(\mathbf{v}(s)) + \mathbf{F}_{ext}(\mathbf{v}(s)) = 0 \quad (2.6)$$

where $\mathbf{F}_{int}(\mathbf{v}(s)) = \alpha \mathbf{v}''(s) - \beta \mathbf{v}''''(s)$ is the internal force to constrain the contour smoothness, and $\mathbf{F}_{ext}(\mathbf{v}(s)) = -\nabla E_{ext}(\mathbf{v}(s))$ is the external force to guide the snake to convergence. $\mathbf{F}_{int}$ and $\mathbf{F}_{ext}$ essentially carry the same meanings as their corresponding energy terms of matching subscripts. In this light, (2.5) and (2.6) can be considered as force-balance equations as the net force is equal to zero. “Forces” are used to describe $\mathbf{F}_{int}$ and $\mathbf{F}_{ext}$ because both terms consist exclusively of vector quantities after a gradient operation is performed on the scalar E_{snake} in (2.5). (If curve $\mathbf{v}(s)$ is written in terms of its spatial coordinates $[x(s), y(s)]$, the functional derivative in 2.5 is equivalent to computing the gradient of E_{snake} with respect to x and y). Essentially, snake evolution starts with placing an initial curve on the image. The locations of the curve are subsequently updated through minimizing (2.4). A converged snake not only minimizes the total energy, it also satisfies the equilibrium between the internal and external forces.

The check and balance between E_{int} and E_{ext} (or $\mathbf{F}_{int}$ and $\mathbf{F}_{ext}$) is required for the implementation of the snake. Without E_{ext} , snake simply cannot evolve. Further, minimization of E_{int} alone will lead to zero energy, or the collapse of snake to a single point. Contrarily, without E_{int} snake becomes overly reactive to high-frequency

components in the image (e.g. image noise) and frequently takes on erroneous shapes that are uncharacteristic to natural objects. The force-balance equation keeps the snake at an equilibrium whereby its free movement is constrained by the overall smoothness of the snake, allowing it to not only localize the shape, but also do so in a more accurate and natural way. Therefore, the force-balance equation in (2.6) captures the essentials of the snake algorithm.

2.1.3 Numerical implementation of active contours

Despite the sound theory behind the force-balance equation representing the snake, a closed form solution to (2.5) or (2.6) is generally unattainable [14]. Thus, we resort to numerical methods. The Euler-Lagrange equation (2.5) is first approximated by a discrete representation. Gradient descent method is then applied to the discrete equation to find a minimizer.

To approximate (2.5), the first step is to discretize the continuous snake $\mathbf{v}(s)$. We replace the continuous curve length parameter s with indices $i \in \{0, 1, 2, \dots, n-1\}$ where n is the total number of discrete vertices on the snake. These discrete vertices are also called “snaxels”. The snaxels can simply be written as $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots, \mathbf{v}_{n-1}\}$.

The next step is to discretize (2.5). After approximating the 2nd and 4th order derivative terms in (2.5) by finite differences of neighboring snaxels, the discrete representation of (2.5) will take the form:

$$\begin{aligned}
& \alpha_i(\mathbf{v}_i - \mathbf{v}_{i-1}) - \alpha_{i+1}(\mathbf{v}_i - \mathbf{v}_{i+1}) \\
& + \beta_{i-1}(\mathbf{v}_{i-2} - 2\mathbf{v}_{i-1} + \mathbf{v}_i) \\
& - 2\beta_i(\mathbf{v}_{i-1} - 2\mathbf{v}_i + \mathbf{v}_{i+1}) \\
& + \beta_{i+1}(\mathbf{v}_i - 2\mathbf{v}_{i+1} + \mathbf{v}_{i+2}) \\
& - (\mathbf{F}_x(\mathbf{v}_i), \mathbf{F}_y(\mathbf{v}_i)) = 0
\end{aligned} \tag{2.7}$$

where $(\mathbf{F}_x(\mathbf{v}_i), \mathbf{F}_y(\mathbf{v}_i))$ is the vector notation of the external image force at snaxel $\mathbf{v}_i$. If we substitute $\mathbf{v}_i$ with its vector notation (X_i, Y_i) , and rewrite (2.7) in matrix form, (2.7) can be considerably simplified to:

$$\begin{aligned}
AX + \mathbf{F}_x(x, y) &= 0 \\
AY + \mathbf{F}_y(x, y) &= 0
\end{aligned} \tag{2.8}$$

The matrix A in (2.8) is a penta-diagonal structure of $n \times n$ size, and it specifies all of the coefficients for the terms in (2.8) that are associated with the internal snake force. In other words, the coefficients that constrain the first and second order smoothness for the snake are all included in A . The coordinates of all snaxels are specified in the X and Y matrices. $\mathbf{F}_x(x, y)$ and $\mathbf{F}_y(x, y)$ again are components of the external image force. Gradient descent is then used to search for X and Y configurations such that they satisfy the force-balance condition of (2.8).

Gradient descent is an optimization method that uses the gradient of a function to find the minimizer for the function. The premise of gradient descent is based the fact that the gradient of a real-valued differentiable function g at x_0 , denoted $\nabla g(x_0)$, always points in the direction of the maximum rate of increase of g at x_0 , which can be proved by the Cauchy-Schwarz inequality [21]. The negative gradient, $-\nabla g(x_0)$, thus, marks the

direction of the maximum rate of decrease of g at x_0 . To find the location where the minimum of g occurs, it would then make sense that an algorithm searches in the direction given by $-\nabla g(x_0)$. Gradient descent iteratively moves the search location x^τ along the direction of $-\nabla g(x^\tau)$, as given in the following equation.

$$\frac{x^{\tau+1} - x^\tau}{\gamma} = -\nabla g(x^\tau) \quad (2.9)$$

where τ denotes the iteration number, and γ is a small scalar constant called the step size, and g is called the objective function. For example, if we are at point x^1 on the surface of g , to look for the minimum of g , we would start at x^1 and move by an amount of $-\gamma \nabla g(x^1)$ to the next point x^2 . By the merit of the negative gradient, we know that $g(x^2) < g(x^1)$, and thus we are indeed finding a lesser value of g . We apply the same technique to locate subsequent points x^3, x^4 , etc. If at some point x^τ , $\nabla g(x^\tau)$ becomes 0, we then have reached the local minimum point of g . In this case, (2.9) would reveal $x^{\tau+1} = x^\tau$, and the gradient descent search terminates. Practically, however, the stopping condition $\nabla g(x^\tau) = 0$ is very rarely realized in numerical calculations. A more suitable stopping condition is adapted, which is shown in (2.10). If the difference in the values of g between two consecutive gradient descent iterations falls below a predefined threshold ε , the search then terminates.

$$|g(x^{\tau+1}) - g(x^\tau)| < \varepsilon \quad (2.10)$$

The gradient descent is applied to the snake. The objective function which we wish to minimize is the total snake energy E_{snake} (2.4). Its gradient with respect to snake positions, $\nabla E_{\text{snake}}(X, Y) = \left(\frac{\partial E_{\text{snake}}}{\partial X}, \frac{\partial E_{\text{snake}}}{\partial Y} \right)$, is solved by the Euler-Lagrange Equation (2.5) and the numerical approximation of (2.5) given in (2.8). To apply gradient descent

to (2.8), we introduce iteration steps to all of the variables. Then we set the right-hand side of (2.8) equal to the left-hand side of (2.9) with a sign change. The pair of equations will take the forms of the following:

$$\begin{aligned} AX^{\tau+1} + \mathbf{F}_x(x^\tau, y^\tau) &= -\frac{X^{\tau+1} - X^\tau}{\gamma} \\ AY^{\tau+1} + \mathbf{F}_y(x^\tau, y^\tau) &= -\frac{Y^{\tau+1} - Y^\tau}{\gamma} \end{aligned} \quad (2.11)$$

After applying matrix inversion of A and rearranging (2.11), we can finally derive a pair of iterative equations for snake evolution:

$$\begin{aligned} X^{\tau+1} &= (I_n + \gamma A)^{-1}(X^\tau - \gamma \mathbf{F}_x(x^\tau, y^\tau)) \\ Y^{\tau+1} &= (I_n + \gamma A)^{-1}(Y^\tau - \gamma \mathbf{F}_y(x^\tau, y^\tau)) \end{aligned} \quad (2.12)$$

I_n is an $n \times n$ sized identity matrix. The snake algorithm requires initial placement of a contour near the image feature, which is not a trivial task and remains to be an open area of research [22]. Following the initial placement, the snaxels are subsequently moved according to (2.12). When the movement of a snaxel from two consecutive iterations becomes negligibly small as what was given in (2.10), the iteration for that snaxel completes. The final positions of the snaxels balance two conditions: 1) they correspond to locations of maximum image gradient force; 2) the contour drawn out by the snaxels is smooth.

2.2 External Force Models for the Active Contour

The Kass *et al* snake [9] uses image gradient force to attract the snake towards edge features. However, this gradient force is extremely limited in terms of providing the adequate capture range for the snake, causing the performance of the snake to suffer. A large volume of research has subsequently aimed at improving the external force model for the snake. This section reviews several common force models in the literature, and bridges a connection to the MVFC model to be introduced in Chapter 3.

Algorithmic implementations for the force models to be discussed in Subsections 2.2.1-2.2.4 are provided by authors of [23].

2.2.1 Image gradient forces

The gradient force employed in the Kass *et al* snake is given in (2.1). It is also addressed as the traditional image force in the literature. It is computed by evaluating the gradient of edge features, and the force strength is expressed by the resultant gradient magnitude.

One major drawback of the traditional force is that it is unable to reach very far away from the feature. Consider a part of the image that contains no features. If the snake is initialized at this featureless region, it may not be reached by gradient forces, and thus no external force can act on the snake to guide it to the feature. Illustrated in Fig. 2-2, snake is initialized off-centered from a “U” shaped feature. The left part of the snake that was farther away from the feature was outside the range of image gradient forces, and thus the left part of the snake failed to evolve towards the “U” shape.

Figure 2-2: Example of image gradient forces. (a) A gradient force field is computed for “U” and it has limited force spread; (b) failed segmentation of the “U” shape. (Dashed contour: initial snake; solid contour: final snake.)

2.2.2 Forces by distance transform

Forces achieved by distance transform improve from traditional forces in that they can have an unlimited range. The caveat is that the distance transform only works for binary images [14]. To derive the distance potential forces from a binary image, a distance map is first constructed from the image. For every pixel in the image, the distance transform finds a distance between the said pixel and the nearest non-zero valued neighboring pixel, as expressed in (2.13). If we assume that a pixel of value “0” represents the background and a “1”-valued pixel represents the feature in the image, the distance map assigns a value to every pixel based on the distance between that pixel and the closest feature found near it.

$$D(x, y) = \min_{(p, q) \in \{(a, b): I(a, b) = 1\}} \sqrt{(x - p)^2 + (y - q)^2} \quad (2.13)$$

The distance potential forces are then created from $D(x, y)$ by evaluating $-\nabla(D(x, y))$.

In the following example, snake driven by distance potential forces was used to segment the binary “U” shape from an input image. Fig. 2-3-(a) displays the distance map built from the binary “U”, where darker values correspond to closer distances. The distance potential force field, shown in Fig. 2-3-(c), spreads to the entire image, significantly improving the force range achieved by traditional image forces. The omnipresent force field greatly facilitates active contour initialization as the initial contour can be placed relatively far away from the feature yet still be attracted to the feature. However, distance potential forces are unable to guide the snake into a concave region, as shown in Fig. 2-3-(b), due to a lack of forces that point directly into the concavity being produced by the method.

Figure 2-3: (a) Distance map of a binary “U” shape; (b) failed segmentation of the “U”. (Dashed contour: initial snake; solid contour: final snake); (c) distance potential forces computed for “U”.

Limitations of the distance transform also include noise sensitivity (consider the addition of one “1”-valued pixel can alter the values assigned to all other pixels in the distance map), and the requirement of binary images.

2.2.3 Balloon forces

Cohen and Cohen introduced the balloon force model [20, 24] to improve the performance of the snake in terms of increased capture range, better noise tolerance and the ability to converge to concave boundaries.

The proposed balloon force model is formulated in (2.14). It adds to the image gradient force another force, termed the pressure force, to make the total force more dynamic.

$$\mathbf{F} = k_1 \mathbf{n}(s) - k_2 \frac{\nabla(|\nabla I(x, y)|^2)}{|\nabla(|\nabla I(x, y)|^2)|} \quad (2.14)$$

In (2.14), $\mathbf{n}(s)$ is a unit vector pointing in the normal direction to the snake, acting to either push the snake outward (inflating the balloon) or push it inward (deflating the balloon). Scalar coefficients k_1 and k_2 control the relative importance of the two forces. The sign of k_1 also controls the direction of the pressure force applied on the snake.

The incorporation of pressure forces extends the external force range for the snake and gives the snake greater freedom of motion. Essentially, when snake is initialized at featureless regions where image gradient forces would become unavailable, it can still be guided to the feature by the pressure forces. For this reason, the pressure forces can push the snake into concave regions where the traditional method has failed. Furthermore, the pressure forces can counteract small amounts of undesired image gradient forces such as those created by a spurious edge, allowing snake to move pass noisy bits of data.

The main disadvantage of the balloon force model lies in the difficulty of stopping the snake at the desired edges. Delicate balance between the inflation forces and the edge gradient forces must be maintained. In practice, k_2 must be set greater than k_1 so that the edge force can check the inflation force at the actual object boundaries. However, if k_2 is too large compared to k_1 snake will stop early at the weaker edges or noise. Additionally, the direction of the pressure forces must be specified in advance, which demands foreknowledge about the image and careful snake initialization.

The following example illustrates segmentation of a “room” shape using balloon forces. Despite of limited range of the traditional gradient force, shown in Fig. 2-4-(a), under the influence of the inflation force snake still successfully moved towards the

feature. However, the inflation force overpowered snake motion and caused it to leak outside the small gaps on the object boundary, as shown in Fig 2-4-(b).

Figure 2-4: Segmentation with balloon forces. (a) Image gradient forces for “room”. (b) failed segmentation of “room” when inflation forces were added to the external force. (Dashed contour: initial snake; solid contour: final snake.)

2.2.4 Gradient Vector Flow model

A major improvement of the snake came with the Gradient Vector Flow (GVF) force model, introduced by Xu and Prince [25, 26]. GVF diffuses the image gradient forces generated by the traditional method using a system of partial differential equations, resulting in a significantly increased force range and spatial convergence to boundaries at concave regions. The result of GVF is also consistent as the model does not rely on heuristic parameters as in the case of the balloon force model.

The GVF force field, written as $\mathbf{v}(x, y) = [u(x, y), v(x, y)]$, is designed to minimize an energy functional given in (2.15). The GVF forces are obtained by applying

variational minimization to (2.15) and solving the resulted Euler-Lagrange equations numerically. (The steps are identical to how snake is solved from the functional of total snake energy.)

$$E_{GVF}(u, v) = \iint \mu \left(\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dx dy \quad (2.15)$$

The rationale of the GVF force field, $\mathbf{v}$, is to extend the image gradient forces into featureless regions by diffusion (identical to e.g. heat diffusion), while also preserving the gradient forces at the image features. In (2.15), the first term in the integrand consists of the sum of diffusion terms of $\mathbf{v}$; the second term in the integrand conditions $\mathbf{v}$ at locations of the edge features. (Constant μ weights in on how strong the diffusion forces will become.) Notation f denotes an edge map of the image, which simply measures the edge strength across the image and is computed by $f = |\nabla I(x, y)|^2$. The image gradient forces are then equal to ∇f . Essentially, at featureless regions where $|\nabla f|$ is small, $\mathbf{v}$ will consist primarily of diffusion forces because (2.15) will be dominated by the sum of 2nd order partial derivatives of $\mathbf{v}$. Minimizing (2.15) will force $\mathbf{v}$ to become very smooth. At the image features, however, $|\nabla f|$ reaches maximum and (2.15) will then be dominated by the image gradient forces. The minimum solution will then be achieved at $\mathbf{v} = \nabla f$, when $\mathbf{v}$ is set equal to the edge forces.

GFV outperforms previous snake external force models by the robustness of the force field it generates. Illustrated in Fig. 2.5, GVF force field extends far beyond the image feature, which provides excellent capture range for the snake to reach a target. The diffusion forces of GVF also allow the snake to reach concave boundaries.

Figure 2-5: Segmentation of a “U” shape with GVF snake. (a) External force field produced by GVF; (b) GVF snake successfully delineates the concave boundary of “U”. (Dashed contour: initial snake; solid contour: final snake.)

GVF is computationally expensive, which becomes its main drawback. The GVF forces are solved iteratively by numerical methods. Algorithmically, convergence to the functional minimum requires the specification of a total iteration number, which also directly influences the range of the final GVF force field. For an $N \times N$ image, it requires a full scan of every pixel location to update the external force field, resulting $O(N^2)$ in computation complexity. This cost is, however, repeated for every GVF iteration. Typically the iteration number is set to be equal to N [26], therefore raising the total computational cost of GVF to $O(N^3)$ [27]. The cost of GVF can become prohibitive when we intend to segment large images.

In addition, it is shown that the GVF-driven snake is sensitive to image noise and clutter [27].

2.2.5 Vector Field Convolution model

Li and Acton introduced a new external force model for the snake termed Vector Field Convolution (VFC) [27, 28]. VFC simulates the vector diffusion of GVF by way of a one-step convolution that can be implemented in the Fourier domain, considerably lowering the computational cost of VFC from that of GVF.

Expressed in (2.16), VFC produces a force field by convolving an image edge map $f(x, y)$ with a pre-configured kernel of vector forces $\mathbf{K}(x, y)$:

$$\mathbf{F}_{VFC}(x, y) = f(x, y) * \mathbf{K}(x, y) \quad (2.16)$$

The vector field kernel (VFK), $\mathbf{K}(x, y)$, is configured from reshaping a unit-length force field, denoted $\mathbf{n}(x, y)$, by a scalar function, $m(x, y)$, where the former gives the orientations of the forces and the latter specifies the force magnitude:

$$\mathbf{K}(x, y) = m(x, y)\mathbf{n}(x, y) \quad (2.17)$$

The unit-length force field is defined in (2.18). Essentially, it contains unit vector forces at every location and the forces always point towards the center of the field.

$$\mathbf{n}(x, y) = \left(-\frac{x}{\sqrt{x^2 + y^2}}, -\frac{y}{\sqrt{x^2 + y^2}} \right) \quad (2.18)$$

Two different magnitude functions for $\mathbf{K}(x, y)$ were proposed by the authors, formulated in (2.19) and (2.20):

$$m_1(x, y) = (r + \epsilon)^{-\gamma} \quad (2.19)$$

$$m_2(x, y) = e^{-r^2/\zeta^2} \quad (2.20)$$

where r in both formulae is the distance from a point (x, y) on the function to its origin, defined at the center point of each function. Parameters γ in (2.19) and ζ in (2.20) each

controls the rate of decay of the respective function; ϵ in (2.19) is a small constant (taking the value of 10^{-8} for example) to prevent division by zero at the origin.

Fig. 2.6 gives is graphical representation that summarizes the steps described above to construct VFK.

Figure 2-6: Configuration of a Vector Field Kernel (VFK). A magnitude function (2.19 or 2.20) is multiplied to a unit vector field kernel to create the VFK.

Both magnitude functions draw inspirations from physical phenomenon. m_1 is inspired by Newton's law of universal gravity, which is a special case of m_1 after setting $\gamma = 2$. m_2 represents Gaussian decay. The magnitude functions are chosen because of their isotropic and strictly-decreasing properties. Both functions are isotopic, meaning that points located at equal distances away from the function center will have the same

magnitudes. The isotropic property is desired because it prevents the diffusion of image forces from being biased towards any particular angle. Next, both functions are also strictly decreasing from their centers. As a result, the configured VFK will have the largest force magnitudes near its center and the rest of the forces will gradually diminish as their distances to the center increase. This coincides with the consideration that an image feature should exert the strongest attraction to a snaxel located closest to it over those located farther away.

The VFK is convolved with an image edge map to produce the final external forces for the snake. Illustrated in the following example, VFC can lead to similar segmentation performance for the snake as GVF.

Figure 2-7: Segmentation of a “U” shape with VFC snake. (a) External force field produced by VFC; (b) VFC snake successfully segments “U”. (Dashed contour: initial snake; solid contour: final snake.)

The main improvements VFC achieved over GVF include considerably lowered computational cost and better performance when impulse noise is present in images.

VFC can be optimized when it is implemented in the frequency domain via the Fast Fourier Transform (FFT). The computational complexity of VFC is thus governed by the speed of the 2D FFT algorithm, known to be $O(N^2 \log(N))$ for an input signal of $N \times N$ size. The speed of VFC is thus more favorable than GVF particularly when large images are involved. It is demonstrated that VFC is 3 to 10 times faster than GVF when external forces over a set of similar ranges are produced by two methods.

Secondly, by the merit of convolution, which computes an averaged response over a fixed area of the input signal determined by the convolution kernel, VFC can smooth away small and isolated impulse noises, achieving better segmentation results than GVF when such noises are present in the image. However, this averaging effect of the convolution has also fundamentally limited the performances of VFC, as will be discussed in detail in the next chapter, which has then prompted the development of the multi-scale VFC algorithm.

3. Multi-scale Vector Field Convolution

3.1 Scale-space Representation

Multi-scale VFC is inspired by scale-space theory. Scale-space provides a basic framework for representing signals at varying scales of observation. In the context of 2D signals, such as images, scale-space provides a way to model image structures as if they were viewed at different distances. Because objects viewed at different distances may have different visual appearances and thus affect a machine's interpretation of what they may be, the viewing distance, or scale, is highly relevant in automatic object detection or recognition tasks. First introduced by Witkin [29], the theory of scale-space has been extended [30, 31] and adapted to an overwhelmingly large number of algorithms concerning automatic feature detection and segmentation [32-37].

The scale-space is typically realized through linear filtering of an image with a family of low-passing filters of increasing widths to produce a series of incrementally blurred versions of the original image. The blurring effect simulates observing the image from the closest distance to progressively farther distances. While viewed at the closest range, every line and corner in the original image is preserved; as the viewing range increases, more and more fine features in the image are smoothed away and only pixels of slower-varying intensity values remain. These pixels belong to more elemental shapes and structures in the image.

Mathematically, a scale-space may be expressed using:

$$F(x, y; t) = f(x, y) * g(x, y; t) \quad (3.1)$$

where $f(x, y)$ represents the original image, “ $*$ ” represents convolution, and $g(x, y; t)$ is a 2D low-pass filter kernel of scale t . Following convolution, smoothed versions of the original image are organized in a scale-space volume $F(x, y; t)$, where every 2D slice in this volume represents one version of the image of a different level of smoothing. The convolution kernel with scale t (the scale parameter for the scale-space) can be used as an index to track fine-to-coarse progressions of the image structures across the specified scale ranges.

Gaussian functions are the ideal choice for the low-passing function in (3.1) because of its general “well-behavedness” as described by Witkin. Firstly, Gaussian is a symmetric and strictly decreasing function about its mean, resulting in that the weights assigned to the image in convolution decreases smoothly with distance. Secondly, with Gaussian convolution, the scale-space behaves well near the limits of the scale parameter: when the Gaussian kernel width approximates 0, the resultant convolution produces the exact same copy of the input image, i.e. non-blurred image; when the Gaussian kernel width approximates infinity, the convolution yields the DC component of the image, i.e. the mean value of the image. Lastly, it is shown that when convolution is performed with Gaussian kernels of increasing widths, no additional local maxima will be created in the scale-space, which guarantees that no false features (those not already present in the original image) will be created by successive linear filtering. The scale-space employing Gaussian kernels takes the form of the following:

$$F(x, y; t) = f(x, y) * e^{-(x^2+y^2)/t^2} \quad (3.2)$$

where the scale parameter t is the standard deviation of the Gaussian kernel, which describes the width of the kernel.

Viewing vector field convolution (VFC) with regard to the scale-space theory allows for a more comprehensive look at VFC, including its fundamental limitations in segmentation. It is this view that propels the development of MVFC.

3.2 Limitations of VFC

Problems of VFC root in the convolution operation itself. The definition of a scalar convolution in 2D is given in (3.3), where f denotes an input image, h is a 2D kernel that provides weights to scale the image, and g is the output of a weighted sum operation between f and a spatially-reversed and shifted version of h . More specifically, h is first flipped about its origin both in the horizontal and vertical directions and then shifted by x and y amounts respectively in each direction to produce a shifted and spatially-reversed copy h' . Inside the overlapped region between f and h' , entries of f is multiplied to corresponding entries of h' and the results are summed to produce $g(x, y)$. When the shift of (x, y) causes h' to move entirely outside of f , $g(x, y)$ results to zero.

$$g(x, y) = \sum_{a=-\infty}^{+\infty} \sum_{b=-\infty}^{+\infty} f(a, b)h(x - a, y - b) \quad (3.3)$$

VFC replaces the scalar kernel h with a vector kernel $\mathbf{K}$ given in (2.2), and then (2.2) can be fully expressed as below:

$$\mathbf{F}(x, y) = \sum_{a=-\infty}^{+\infty} \sum_{b=-\infty}^{+\infty} f(a, b)\mathbf{K}(x - a, y - b) \quad (3.4)$$

VFC calculates weighted sum in terms of vectors. To demonstrate VFC, an example is included. Shown in Fig. 3.1, an image containing a gray-valued “U” shape is used as the input edge map. White pixels in this map have values of 1; black pixels have values of 0; and gray pixels fall in the range between 0 and 1. The image is convolved with a vector field kernel (VFK) displayed as a square region containing force vectors as blue arrows. The goal is to demonstrate the calculation of a single external force at an arbitrary location marked as (x_0, y_0) using (3.4). Steps are shown below.

A) Assume the origins of both the image and the VFK are defined at their centers.

The VFK is first flipped both left-to-right and up-to-down about its origin, resulting the vectors to be rotated 180° and now all point away from the origin, as shown by the blue arrows.

B) The VFK is shifted x_0 pixels leftward and y_0 pixels upward so that the kernel origin now aligns with (x_0, y_0) point on the image.

C) Only the vectors falling inside the image (vectors selected by the red box) contribute to the convolution. They are weighted by the magnitude of the edge feature and then summed to produce one vector $\mathbf{F}(x_0, y_0)$, marked by the red arrow, and it represents the weighted average of all force vectors inside the effective convolution region.

It is evident from Fig. 3.1 that the part of “U” that overlaps with the VFK falls entirely inside the lower-right quadrant of the rotated VFK where the vectors all point towards the southeast direction. The weighted average of these vectors therefore is also oriented towards southeast, as shown by the arrow in Fig. 3.1. It means that when a snake (active contour) is placed at location (x_0, y_0) it will be pulled towards the “U” shape.

Figure 3-1: Illustration of VFC external force calculated for point (x_0, y_0) . The “U” shape image is the edge map $f(x, y)$. Blue arrows mark a spatially-reversed VFK. The offsets of the VFK are marked in green. The red rectangle outlines an area in which force vectors of the VFK is weighted by the image and summed to produce the external force at (x_0, y_0) , $F(x_0, y_0)$, marked by the red arrow.

It is precisely the averaging effect of convolution that causes problems for VFC. In case of vector summation, force vectors are subject to cancellation or directional bias depending on where the kernel is placed and on the size of the kernel. At several key feature locations the force cancellation or biasing effects may become so critical that the resulted external forces can misguide the snake into incorrect segmentation. Examples of such key feature locations include points of high-curvature or junctions between closely-

spaced features. In addition, the size of the VFK also plays an important role. A larger VFK means that more force vectors will be summed, leading to an inferior force field that is plagued by bias and/or cancelation.

Two examples can serve as simple demonstrations that VFC-driven snake fails to capture 1) an object containing high curvature features and 2) multiple objects separated by a small distance.

The three angles of a triangle are sharp corners and are considered as high-curvature features. Fig. 3.2 (a) illustrates a VFK reversed and shifted to the position of the leftmost angle of the triangle. Inside the valid convolution region (inside the red box) only vectors falling onto the two sides of the angle contribute to the calculation of the external force for the said angle location (all other vectors are zeroed when they fall onto the black region of the image). As a result, the average force vector at this location should point at an angle between the two sides, as marked by the red arrow. This force will prevent the snake from converging at the angle position, as the snake is expected to be. When the snake is initialized outside the triangle, it eventually gets pulled inside the triangle and results segmentation error at the corners. This is shown in Fig. 3.2 (b).

Figure 3-2: Limitation of VFC at corners. (a): schematic of a spatially-reversed VFK centered on the leftmost angle of a “triangle” shape and the external force at that angle point (marked by red arrow). (b): VFC external forces generated for the “triangle” using the VFK shown in (a) and snake segmentation of the “triangle”. (Dotted oval contour is the initial snake and solid red contour is the final segmentation.)

In images that contain multiple objects, VFC may also lead to error in segmentation. If the separation distance between adjacent objects is less than the VFK width, VFK will slide over parts of both objects simultaneously. Consequently, the summed external force is under mutual influences from both objects, and force interference between the two objects will occur. In the next example, two neighboring squares of different sizes are used as test objects. Fig. 3.3 (a) displays a VFK shifted to the center of the large square to compute the external force at its center location. The contributing vectors in the convolution (again the vectors inside the red box) include those laying on all four sides of the large square as well as those laying partially on the small square. The latter group of vectors breaks otherwise a zero net force established by

the symmetry of the VFK and the four sides of the large square about its center. As a result, the external force at the center position is non-zero and is slightly biased towards the direction of the small square, as marked by the red arrow.

In fact, the external forces at nearby locations in the large square all will be biased by the small square. This explains why in Fig. 3.3 (c) the snake initialized at the center of the large square is pulled by the small square and eventually leaks outside of the large square. The same analysis also applies when two squares switch places as shown in Fig. 3.3 (b). The influence the large square exerts on the small square is much stronger than the opposite case. As evident by Fig. 3.3 (c), force distortion inside the small square is much worse than that inside the large square. While the snake for the large square only leaks outside at the object's boundary, the snake for the small square almost completely collapses.

Figure 3-3: Limitation of VFC involving two objects. (a and b): schematics of a spatially-reversed VFK shifted to the centers of two squares of different sizes (c): VFC external forces generated for the image containing two squares using the VFK shown in (a, b) and final snake segmentation of both squares. (Dotted circles are the initial snakes and solid contours demonstrate the final segmentations.)

In light of scale-space, VFC can be viewed as a fixed-scale process due to the use of a fixed-width kernel in generating external forces. As such, VFC applies the same amount of force diffusion at every image location regardless of how the actual image structure may be. This becomes a significant source of error for VFC. On one hand, a large kernel is almost always desired to produce enough force spreading so that the active contour can have an ideal capture range. On the other hand, a large kernel always impedes the accuracy of the active contour due to increased force interference and biasing. By introducing a variable scale to VFC, we show that a large force spread and fidelity of the force can be achieved altogether.

3.3 Multi-scale VFC

Multi-scale VFC, introduced here, incorporates a variable scale parameter in the existing VFC framework, thereby allowing scale selection in building the external force field for the active contour, resulting in a new force field that can adapt to changes in local image structure. Inspired by scale-space, MVFC convolves an image edge map $f(x, y)$ with a bank of vector kernels over a discrete range of scales t :

$$\mathbf{F}_V(x, y; t) = f(x, y) * [m_{norm}(x, y; t) \cdot \mathbf{n}(x, y)] \quad (3.5)$$

where $t \in [t_{min}, t_{max}]$. The vector kernel is constructed in the same way as with VFC by way of multiplication of a magnitude function $m(x, y)$ with a unit vector field $\mathbf{n}(x, y)$. Power-decay (2.19) and Gaussian-decay (2.20) functions proposed in the original VFC remain as viable magnitude functions for MVFC, given the desired isotropic and strictly-decreasing properties of these functions. In essence, MVFC represents a collection of VFCs at varying scales.

The capture range of a MVFC field is explicitly given by the width of the largest VFC kernel deployed in the MVFC filter set, which is $t_{\max}$. The minimum resolution MVFC could achieve is imposed by $t_{\min}$, which directly attributes to the fidelity of the MVFC forces. The total number of scales to construct MVFC is arbitrary, and it is typically determined by the computational cost constraint. MVFC stores derived VFC force fields at the specified scales in a volume $\mathbf{F}_{\mathbf{v}}(x, y; t)$ for which the axis along the depth dimension is the scale axis.

The term m_{norm} in (3.5) represents normalized force magnitude. The VFK is normalized so that the sum of magnitude of all forces in the VFK equals to 1 (L_1 normalization). Normalization is needed in order to compare filter outputs across different scales. The scale parameter is embedded in $m_{\text{norm}}(x, y; t)$, and it controls the width of the VFK. Without normalization, when the scale changes, total weights assigned to the unit VFK also changes. As a result, the external forces produced at the same location but at different scales are uncorrelated and thus cannot be compared on equal footing. Normalization makes it possible to adaptively select an optimal scale, and in turn an optimal external force given the chosen scale, at every location.

Both m_1 and m_2 are isotropic functions. The sum of all weights (L_1 sum) can be computed for each function in closed forms conveniently by taking the integral of each function with respect to the angle and the radius in a cylindrical coordinate system. For the power-decaying function, we consider the special case where the power $\gamma = 1$. Normalized (2.19) and (2.20) are thus given as below:

$$m_{1,norm}(x, y; t) = \frac{(r + \epsilon)^{-1}}{2\pi t - \pi r_0} \quad (3.6)$$

$$m_{2,norm}(x, y; t) = \frac{\exp(-r^2/t^2)}{2t^2} \quad (3.7)$$

Because m_1 approaches $+\infty$ at location $(0, 0)$, the function is truncated at a small radius r_0 in order to approximate the total volume underneath.

The MVFC kernel sets adopting the above-mentioned normalized magnitude functions are shown in Fig. 3.4 and 3.5.

Figure 3-4: Cross-sectional magnitude profiles of power-decaying (3.6) MVFC kernel set. It provides a force spreading range of 100 pixels and consists of 5 scales linearly spaced between $t_{\min} = 10$ pixels and $t_{\max} = 100$ pixels. Scale t is defined explicitly as the width of the function.

Figure 3-5: Cross-sectional magnitude profiles of Gaussian-decaying (3.7) MVFC kernel set. It provides a force spreading range of 100 pixels and consists of 5 scales exponentially spaced between $t_{\min} = 4$ pixels and $t_{\max} = 25$ pixels. Here scale t is equivalent to the standard deviation of the Gaussian. The actual width of the function, hence the force spreading range, is implemented as $4t$.

At this point, the scale selection procedure must be addressed. For every location (x, y) in the image, an optimal scale is chosen such that it corresponds to the maximum normalized VFC force across all scales being produced at that location:

$$t_{opt}(x, y) = \arg \max_{t_{\min} \leq t \leq t_{\max}} |F_V(x, y; t)| \quad (3.8)$$

The MVFC force field is then created by combining the normalized VFC forces selected at each (x, y) into a single force field:

$$\mathbf{F}_{MVFC}(x, y) = \mathbf{F}_V(x, y; t_{opt}) \quad (3.9)$$

An illustration of MVFC is given in Fig. 3.6. The force-scale volume $\mathbf{F}_V(x, y; t)$ is shown graphically as a stack of VFC force fields arranged in the order of increasing scale. In addition, each VFC force field is normalized by the total force magnitude contained in the field. The MVFC algorithm then traverses through every location (x_0, y_0) and searches for an optimal external force for that location using previously described scale selection criterion.

Figure 3-6: Illustration of MVFC. It consists of multiple versions of VFCs, each computed at a particular scale t_i ($i = 1, 2, 3 \dots$) and normalized by the total force magnitude of the field. For a common location (x_0, y_0) , an optimal scale is selected at (x_0, y_0) such that the selected scale corresponds to the maximum normalized VFC force among all the VFC forces computed at the same location. The MVFC force at (x_0, y_0) is then the found maximum VFC force of the optimal scale.

Identical to scale space, MVFC stores a copy of the external forces due to an image feature at every scale. The scale selection metric adaptively builds a new external force field by matching the VFC forces of the most appropriate scale to every image location. In the next chapter, we demonstrate that MVFC resolves a dilemma of VFC, that of extending the image forces to a desired far range while maintaining the fidelity of the forces at key image locations. Both synthetic and real-world examples will be used to support this claim.

4. Results and Analysis

In this section we compare the proposed MVFC with two most widely used external force generation methods: VFC and GVF. Experiments were conducted using both synthetic and real images to evaluate MVFC, VFC and GVF. Images used in these experiments contain high-curvature features. They may also contain multiple targets, or may be corrupted by noise. Evaluations of the three force models took into consideration the force range, integrity of the forces, noise sensitivity and computational cost of each model.

A. Segmentation evaluation criterion

The integrity of external forces by each model is measured by the segmentation accuracy of snakes. To quantify segmentation accuracy, the Average Shortest Distance (ASD) [38] was used to compute the disparity in pixels between a ground truth boundary and a segmented boundary of feature(s) in every image. Formula of ASD is given in (4.1), where $\mathbf{v}^c$ and $\mathbf{v}^t$ denote points on the converged snake boundary and on the ground-truth boundary respectively; q (n) is the number of points on $\mathbf{v}^c$ ($\mathbf{v}^t$). ASD allows for the calculation of error when $q \neq n$, which is the case in all of the experimental results obtained.

$$ASD = \frac{1}{2q} \sqrt{\sum_{i=1}^q \min_{j=1,2,\dots,n} (\|\mathbf{v}^t(s_j) - \mathbf{v}^c(s_i)\|_2)} + \frac{1}{2n} \sqrt{\sum_{j=1}^n \min_{i=1,2,\dots,q} (\|\mathbf{v}^t(s_j) - \mathbf{v}^c(s_i)\|_2)} \quad (4.1)$$

B. Experimental setup

In all experiments, VFC and MVFC force kernels were implemented using the power-decaying magnitude function: $(r+\varepsilon)^{-\gamma}$, where parameters were set as $\gamma = 1$ and $\varepsilon = 10^{-8}$.

4.1 Synthetic Examples

Synthetic images of “Triangle” and “Two squares” previously shown in Chapter 3 were used again to evaluate the convergence ACs driven by VFC and MVFC at high-curvature points and at junctions between neighboring objects.

4.1.1 Corner convergence

A total of 15 scales, linearly spaced between $t_{\min} = 5$ and $t_{\max} = 100$ pixels were used to generate 15 unique VFC kernels, which then were convolved with the feature image to produce the external forces. The results are shown in Fig. 4-1. As scale increases, VFC segmentation error first decreases, reaching a minimum point, and then it increases again. The initial decrease in segmentation error was due to increased force capture range of the VFC fields. However, as the scale continues to increase, the VFC field becomes more ill conditioned due to worsened force biasing, causing the ACs driven by the VFC to miss the high curvature corners of the triangle. The best segmentation result VFC achieved was at scale $t=46$. However, it is in general difficult to know the most appropriate scale in advance. Thus, to choose the optimal scale for VFC is an *ad hoc* task.

Figure 4-1: Segmenting “Triangle” using VFC snakes computed at multiple scales ($t = [5,100]$ pixels). *Top*: segmentation error (ASD) vs. VFC kernel scale. *Bottom*: segmentation results (dotted circles: initial snake; solid contours: final snake; blue arrows: VFC force field).

MVFC external forces were generated from 15 vector kernels that used the same set of scales as given previously for VFCs. Fig. 4-2 shows the segmentation result for MVFC. In contrast to VFC, a large force range and fidelity of forces at three angles are achieved simultaneously by MVFC. The fidelity of forces at the corners led to significantly improved segmentation.

Figure 4-2: Segmenting “Triangle” using MVFC snake (dotted circles: initial snake; solid contours: final snake; blue arrows: MVFC force field).

Table 4.1 summarizes the performances of the best-case VFC and MVFC for segmenting the “Triangle” shape. MVFC produced the most accurate segmentation and the largest possible force field, outperforming VFC in both categories.

	Best-case VFC	MVFC
Min. ASD (pixels)	6.36	0.85
Force range (pixels)	46	100

Table 4-1: Comparisons of the best-case VFC and MVFC in minimum segmentation error and force capture range for “Triangle”.

The following sets of figures demonstrate adaptive scale selection by MVFC. In the experiment, 4 points were picked from the “Triangle” image, shown to the left. A is an angle point. B, C and D are located outside the triangle. Distances from B, C and D to A are in an increasing order. We then keep track of the magnitudes of the normalized external forces computed by MVFC across all scales at these 4 locations and plot them in Fig. 4.3. The scales that gave rise to the maximum force magnitude for each location are labeled in the figure, and they correspond to the optimal scales selected by MVFC at these locations. The scales MVFC assigned to A, B, C and D are proportional to the distances from each point to A.

Fig. 4.4 plots the scales selected by MVFC across the “Triangle” image. MVFC selects larger kernels at image locations farther away from the feature, and smaller kernels near feature boundaries, which meets the goal of multi-scale feature detection. At feature-sparse region, large kernel is desired as it extends image forces into those regions. At feature-dense regions, on the other hand, a small kernel is needed to preserve the fidelity of the external forces around feature points.

Figure 4-3: (a) Three arbitrary locations (B, C, D) were selected from the image with increasing distances from the corner point A ($AB = 23$ pixels, $AC = 52$ pixels and $AD = 87$ pixels). Normalized external force magnitude across all scales at A, B, C and D are plotted. MVFC finds an optimal scale for each point where the normalized force magnitude is at maximum.

Figure 4-4: Optimal scales selected by MVFC across the full image. Colorbar shows the values of scales.

4.1.2. Multiple object segmentation

“Two squares” was used to evaluate multi-object segmentation of VFC and MVFC snakes.

Fig. 4-5 shows the segmentation results using variable-scale VFCs. A total of 15 VFC force fields were generated using 15 different scales. The scales range from $t_{\min}=10$ to $t_{\max} = 125$ pixels with the remaining values taken in fixed intervals between the minimum and the maximum points. Fig. 4-5 demonstrates that VFCs with scales less than 60 pixels segmented the small square more accurately because the range for forces produced by these VFCs did not reach the large square. As scale increases, VFC forces finally reached the large square; however, force bias and interference between two neighboring objects also became stronger. As a result, segmentation for both objects suffered. For the large square, AC partially leaked outside its boundary. For the small square, the leakage of AC was more severe and in extreme cases the AC was seen to have collapsed to a point.

The best segmentation achieved by VFC is at scale $t=57$ pixels; however, this optimal choice of scale for VFC cannot be known in advance. VFC failed to segment both object correctly at the same time. The difference in size between the two objects and their demand for individual treatment in terms of scale selection render a fixed-scale method such as VFC inadequate for multiple object segmentation.

Figure 4-5: Segmenting “Two squares” with VFC snakes at multiple scales ($t = [5, 125]$ pixels). *Top*: segmentation errors (ASDs) computed separately for large (*LSq*) and small (*SSq*) squares and the sum of errors vs. VFC kernel scale. *Bottom*: segmentation results (dotted circles: initial snake; thin contours: snake evolutions; bold contours: final snake).

The MVFC force field was constructed using the same range of scales that produced the VFCs. Fig. 4-6 shows the segmentation result by MVFC snakes. MVFC produced a large force range and achieved accurate segmentation for both squares at the same time.

The initial ACs were shown as dotted circles at the center of each square. Also depicted in Fig. 4-6, the ACs initiated from the centers of both squares evolved with uniform speed in every direction, creating a ripple of intermediate contours all symmetric about the center. This indicates that the MVFC force field is free of directional force biasing and interferences due to neighboring objects.

Figure 4-6: Segmenting “Two squares” using MVFC snake. (Dotted circles: initial snake; thin contours: snake evolutions; bold contours: final snake).

Table 4.2 summarizes the performance of VFC and MVFC snakes for the “Two Squares” experiment. Once again, MVFC not only yielded better segmentation but also a larger force capture range.

	VFC	MVFC
Min. ASD (pixels)	4.39	0.85
Force range (pixels)	57	125

Table 4-2: Comparisons of the best-case VFC and MVFC in segmentation error and force capture range for “Two squares”.

4.2 Segmentation using Weizmann [39, 40] Database

A set of shapes were chosen from the Weizmann segmentation evaluation database. These objects chosen contain multiple high-curvature features, creating challenges for accurate shape recovery. MVFC, VFC, and GVF forces are generated from the same edge maps derived from these images via the Canny edge detection algorithm. The goal is to evaluate these three force methods in terms of fidelity at high curvature points.

The range of image forces produced from three methods are held at similar values in the simulation for each object to give fair comparison to segmentation in terms of convergence at the high-curvature features. (The maximum scale of MVFC is set equal to the fixed-scale VFC, and thus the capture range of both forces are identical.)

As shown in Fig. 4-10 and Table 4-1, MVFC and GVF-driven snakes yielded comparable results, both accurately captured the intended objects. VFC gave the poorest results among the three methods, failing to capture key image points due to fixed-scale convolution. With the same initial condition, MVFC-driven snake significantly outperformed VFC-snake.

Figure 4-7: Segmentation results using MVFC, VFC and GVF-snake for four objects containing high-curvature features. (Dashed green contours: initial snake; solid red contours: final snake)

	MVFC	VFC	GVF
LEAF	1.08	1.61	1.20
FISH	1.56	10.5	1.52
STAR	1.02	10.6	1.01
BIRD	1.20	77.3	1.21

Table 4-3: Segmentation accuracy (measured in ASD in pixels) for objects in Fig. 4.7. (Bold font indicates minimum error.)

Next, to evaluate how image noise may impact the external force fields and in turn segmentation, Gaussian noise ($\sigma = 0.2$, $\mu = 0$) is added to the images previously shown. Fig. 4-8 and Table 2-4 give corresponding results for segmentation in the presence of additive Gaussian noise, showing that MVFC is more tolerate of image noise than GVF.

Figure 4-8: Segmentation results using MVFC, VFC and GVF-snake for the same objects in Fig. 4-7 but with additive noise. (Dashed green contours: initial snake; solid red contours: final snake)

	MVFC	VFC	GVF
LEAF	4.29	4.07	3.56
FISH	14.2	14.4	22.8
STAR	12.6	28.4	35.0

Table 4-4: Segmentation accuracy (measured in ASD in pixels) for noisy objects in Fig. 4-8. (Bold font indicates minimum error.)

4.3 Computational Cost Analysis

The VFC kernel can be written in terms of its vector components: $\mathbf{K}(x, y) = [u_k(x, y), v_k(x, y)]$. The vector convolution is computed by two scalar convolutions, as shown below:

$$\mathbf{F}_{VFC}(x, y) = [f(x, y) * u_k(x, y), f(x, y) * v_k(x, y)]$$

To implement VFC, $\mathbf{K}(x, y)$ is treated as a complex number having a real part and an imaginary part. This treatment allows vector convolution to be computed in the same way as complex numbers in the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) algorithms.

Computational cost is compared between MVFC and GVF. Because MVFC calculates VFC at n scales, the cost for MVFC is compounded by: 1) the size of the convolution kernel and 2) number of scales. The computational complexity of MVFC is, in theory, n times that of VFC, or $O(nN^2 \log N)$, which still is favorable when compared to GVF having computational complexity of $O(N^3)$ [27, 28]. The difference in the computational time between MVFC and GVF will only widen with increase in image size.

An impulse image of size $N \times N$ was used to test the force computation of MVFC, GVF, and VFC, where N varied from 16 to 640 pixels. Fig. 4-9 shows that MVFC compares favorably against GVF in terms of computational speed in three separate tests with variable scale configurations. For a 640×640 image, MVFC demonstrated to be between 3 to 10 times faster than GVF.

Figure 4-9: Computational cost for image force generation using GVF and 3 different MVFC filter sets (n is number of scales per filter set) for an $N \times N$ image.

5 Conclusions and Discussion

An extension to the VFC method of generating external forces to guide a parametric active contour is developed in this thesis. This extension provides a multi-scale framework that brings benefits of fidelity at corners (and other high curvature features) and extends force capture range that lessens the dependence on accurate active contour initialization. The method fares better than the fixed-resolution version in multi-object segmentation as the adaptive scale parameter avoids artifact prone interaction between objects. MVFC operates at a reduced computational cost, as compared with GVF. MVFC also demonstrates noise tolerance.

Bibliography

- [1] M. A. Costa and D. I. Simon, "Molecular basis of restenosis and drug-eluting stents," *Circulation*, vol. 111, pp. 2257-2273, 2005.
- [2] G. K. Owens, M. S. Kumar, and B. R. Wamhoff, "Molecular regulation of vascular smooth muscle cell differentiation in development and disease," *Physiological Review*, vol. 84, pp. 767-801, 2004.
- [3] B. R. Wamhoff, K. R. Lynch, T. L. MacDonald, and G. K. Owens, "Sphingosine-1-phosphate receptor subtypes differentially regulate smooth muscle cell phenotype," *Arteriosclerosis, Thrombosis and Vascular Biology*, vol. 28, pp. 1454-1461, 2008.
- [4] S. T. Acton, C. Yang, J. A. Hossack, and B. R. Wamhoff, "Poisson inverse gradient approach to vascular myocyte detection and segmentation," presented at the IEEE International Symposium on Biomedical Imaging, Boston, MA, 2009.
- [5] (Apr. 10, 2012). *National Radio Astronomy Observatry EVLA Demonstration Science*. Available: <http://science.nrao.edu/facilities/evla/early-science/demoscience>
- [6] J. T. Pottage, D. R. Flower, and S. L. Davis, "The rotational excitation of methanol by para-hydrogen," *Monthly Notices of the Royal Astronomical Society*, vol. 352, pp. 39-43, 2004.
- [7] L. A. Zapata, J. Schmid-Burgk, and K. M. Menten, "Orion KL: the hot core that is not a "hot core", " *Astronomy & Astrophysics*, vol. 529, 2011.
- [8] B. Harris, C. Yang, A. Remijan, C. Brogan, K. Lehmann, and B. Pate, "Cyber exploration of interstella medium: a new age of broadband data cube analysis," poster presented at: 105th meeting of the American Astronomy Society, Austin, TX, 2012.
- [9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988.
- [10] T. McInerney and D. Terzopoulos, "Topology adaptive deformable surfaces for medical image volume segmentation," *IEEE Transactions on Medical Imaging*, vol. 18, Oct 1999.
- [11] T. McInerney and D. Terzopoulos, "T-snakes: topology adaptive snakes," *Medical Image Analysis*, vol. 4, pp. 73-91, 2000.
- [12] J.-O. Lachaud and A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction," *Medical Image Analysis*, vol. 3, pp. 187-207, 1999.
- [13] M. Gastaud, M. Barlaud, and G. Aubert, "Combining shape prior and statistical features for active contour segmentation," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 14, pp. 726-734, May 2004.
- [14] S. T. Acton and N. Ray, *Biomedical Image Anslysis: Segmentation*, 1 ed. San Rafael, CA: Morgan & Claypool, 2009.

- [15] N. Ray and S. T. Acton, "Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 1466-1478, Dec 2004.
- [16] N. Ray, S. T. Acton, and K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 1222-1235, Oct 2002.
- [17] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 266-280, Mar 2000.
- [18] S. T. Acton and N. Ray, *Biomedical Image Analysis: Tracking*, 1 ed. San Rafael, CA: Morgan and Claypool, 2006.
- [19] (Apr. 10, 2012). *Electric charge*. Available: http://en.wikipedia.org/wiki/Electric_charge
- [20] L. D. Cohen, "On Active Contour Models and Balloons," *Cvgip-Image Understanding*, vol. 53, pp. 211-218, Mar 1991.
- [21] E. K. P. Chong and S. H. Zak, *An Introduction To Optimization*, 2 ed. New York, NY: John Wiley & Sons, 2001.
- [22] B. Li and S. T. Acton, "Automatic active contour initialization via poisson inverse gradient," *IEEE Transactions on Image Processing*, vol. 17, pp. 1406-1420, Aug 2008.
- [23] C. Xu and J. L. Prince. (April 10, 2012). *Active contours, deformable models, and gradient vector flow*. Available: <http://www.iacel.ece.jhu.edu/static/gvf/>
- [24] L. D. 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*, vol. 15, pp. 1131-1147, Nov 1993.
- [25] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours," *Signal Processing*, vol. 71, pp. 131-139, Dec 1998.
- [26] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, pp. 359-369, Mar 1998.
- [27] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," *IEEE Transactions on Image Processing*, vol. 16, pp. 2096-2106, Aug 2007.
- [28] B. Li and S. T. Acton, "Vector field convolution for image segmentation using snakes," presented at the IEEE International Conference on Image Processing, Atlanta, GA, 2006.
- [29] A. P. Witkin, "Scale-space filtering," in *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, 1983, pp. 1019-1022.

- [30] E. Pauwels, L. v. Gool, P. Fiddelaers, and T. Moons, "An extended class of scale-invariant and recursive scale space filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 691-701, 1995.
- [31] T. Lindeberg, "Scale-space theory: a basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, pp. 224-270, 1994.
- [32] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, p. 465, 1996.
- [33] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, pp. 77-116, 1998.
- [34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, p. 91, 2004.
- [35] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.
- [36] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proceedings of Medical Image Computing and Computer-assisted Intervention*, 1998, pp. 130-137.
- [37] J. Staal, M. D. Abramoff, M. Niemeijer, and M. A. Viergever, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 501-509, Apr 2004.
- [38] A. K. Mishra, P. W. Fieguth, and D. S. Clausi, "Decoupled active contour (DAC) for boundary detection," *IEEE Transactions on Image Processing*, vol. 33, pp. 310-324, Feb 2011.
- [39] S. Alpert, M. Galun, T. Bastri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
- [40] (Apr. 10 2012). *Weizmann Segmentation evaluation database*. Available: http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/index.html