Fast Image Segmentation Using Variational Optimization Methods With Edge Detector

Master Thesis in
Applied and Computational Mathematics

Mary Gerina Stanislaus

Department of Mathematics
University of Bergen
Norway

June 2014

Acknowledgement

First of all I want to thank my supervisor professor Xue-Cheng Tai for his mathematical advices and his ideas. In addition, I thank my fantastic office-mate Mari for making our office the place to be in the math department. I enjoyed our conversations, liked to suffer together before exams and appreciated that you helped me when I was frustrated with my work.

Furthermore, I am grateful to have a lovely family and fabulous friends which supported me emotionally in my student years and reminded me that there exists a world outside of university. Especially, I want to thank all people from π-happy which taught me that math people are also able to have fun. The university would not be the same without you. Additional, I am thankful that Mauricio gave me a deeper understanding of the English language and the purpose of mathematical writing.

Finally, I want to thank my Christian for always being there for me.

Mary Gerina Stanislaus,
Bergen, June 2014.

Abstract

In this work, we apply techniques in variational optimization to image segmentation. We study three different segmentation models: one is based on the active contour method, the second is based on a piecewise constant level set method, and the last uses a continuous max-flow min-cut model. We obtain significantly better segmentation results in the first and the third model by including an experimental edge detector. The first model is a special case of the minimal partition problem, the second model uses discontinuities of piecewise constant level set functions to represent interfaces between the region of interest and the background, and the third model uses a spatially continuous max-flow min-cut framework which is a very efficient method to segment images. The first two models are non-convex and may contain many local solutions, but the last model is a convex optimization problem and therefore finds the global solution.

Contents

Chapter 1

Introduction

Digital image processing is a branch of applied mathematics used to find order and pattern in digital images, typically two or three dimensional, and is defined as a discipline in which both the input and output of the process are images. This process improves the pictorial information for human interpretation and uses the processed image data e.g. for storage and transmission. Human vision is restricted to the visual band of the electromagnetic (EM) spectrum, while imaging machines can use almost the whole EM spectrum ranging from gamma to radio waves. Therefore, digital image processing covers a wide and varied field of applications which includes ultrasound, MRI, electron microscopy, and computer-generated images. Image restoration, enhancement, recognition, and segmentation are examples of processes in digital image processing. For example, medical doctors are using MRI and CT to diagnose diseases. Furthermore, image processing can obtain three dimensional information by only using two dimensional input data, a method called 2D-3D reconstruction [28].

Image segmentation is one of many image processing tasks. This process is used when the interest is to extract objects from digital images. In medical imaging this process is used to locate tumours and measure tissue volumes, and in automated license plate reading this process is used for traffic monitoring and surveillance systems.

In this thesis we focus on three different models of image segmentation using a variational approach with an experimental edge detector. The preliminaries for this thesis are written in chapter 2. In this chapter we look at some basics in digital image processing and some optimization methods used in forthcoming chapters.

In chapter 3 we study a curve evolution technique, Mumford-Shah segmentation and the zero level set of a continuous function to detect objects in a given image. This model, proposed by Chan and Vese [2], uses active

contours and minimizes an energy functional which is a special case of the minimal partition problem. After formulating the model by level set functions, the associated Euler-Lagrange equation is solved numerically.

The variational approach in chapter 4 is also based on level set method, but in this method the discontinuities of the piecewise constant level set functions are the interfaces. The problem is to minimize an energy functional under a quadratic constraint. This model, by Lie, Lysaker and Tai [3], uses a piecewise constant level set method applied to Mumford-Shah segmentation, where the level set functions only converge to the two values 1 or -1.

The models in chapter 3 and 4 are non-convex and therefore they are not guaranteed to converge to a global minimum. In chapter 5, however, we study another variational approach, proposed by Yuan, Bae, Tai and Boykov [4], using a max-flow and min-cut model in the spatially continuous setting. This model is convex and solves the associated non-convex problem globally by using a fast max-flow based algorithm.

Chapter 2

Preliminaries

In this chapter we first look at some basics in digital image processing and thereafter look at some optimization methods used in this thesis.

2.1 Basics in Digital Image Processing

This section is about some basic concepts in digital imaging, such as the definition of a digital image and pixels, and an introduction to image segmentation by using edge detectors and the level set method.

2.1.1 Digital imaging

An image is defined as a two-dimensional function $f : \Omega \to \mathbb{R}$, where Ω is a domain of $\mathbb{R}^2$ and $f(x, y)$ is either continuous or discrete. In medical imaging three-dimensional images are also used, e.g in 3D tomography, but in this work we focus on 2D images. The input variables x and y are the *spatial* (plane) coordinates, and the output of f at (x, y) is called the intensity of the image at that point. If x, y, and the values at $f(x, y)$ are all finite and discrete, the image is called a *digital image*.

Assume that illumination energy is reflected from an object of a "scene". Then a sensor, for instance a digital camera, is used to transform the reflected energy into a digital image. When the output of the sensor is a continuous image, this image needs to be converted into digital form. This process is done by digitizing the coordinates and the intensity values, which is also known as *sampling* and *quantization*, respectively. The continuous image can for instance be sampled into a two-dimensional array, $f(x, y)$, with M rows and N columns. Then (x, y) are the discrete coordinates, also known as *pixels*, with integer values $x = 1, \ldots, M$ and $y = 1, \ldots, N$. The number

of intensity levels, denoted by L, is typically an integer power of 2. The discrete intensity values, which are assumed to be equally spaced, are in the interval $[0, L-1]$. Intensities with value zero stands for pure black colour while higher levels of intensity is brighter with $L-1$ being pure white colour. For reasons of simplicity, the interval $[0, L-1]$ is often normalized to $[0, 1]$ by dividing all intensity values by $L-1$. As a consequence, the L intensity values are no longer integers.

Greyscale & Binary Images

A continuous or discrete image, defined by $f : \Omega \to [0, 1]$ for the domain $\Omega \subset \mathbb{R}^2$, which only consists of intensity values, is called a *greyscale image*. If the image is defined by $f : \Omega \to \{0, 1\}$, which means that the image only consists of black and white colours, it is called a *binary image*.

Neighbours of a Pixel

Assume that a pixel $p \in \mathbb{R}^2$ with coordinates (x, y) is given. Then the set of pixels, given by the coordinates

$$(x+1, y),\ (x-1, y),\ (x, y+1),\ (x, y-1),$$

which are the two vertical neighbours and the two horizontal neighbours of p, is denoted by $\mathcal{N}(p)$. The Euclidean distance between the pixel p and its neighbours is 1.

2.1.2 Image Segmentation

An *image segmentation* of an image represented by a spatial region Ω, is the process where Ω is being partitioned into n subregions, $\Omega_1, \ldots, \Omega_n$, such that:

- $\bigcup_{i=1}^{n} \Omega_i = \Omega$,
- Ω_i is a connected set for $i = 1, \ldots, n$,
- the subregions Ω_i and Ω_j are non-overlapping, i.e. $\Omega_i \cap \Omega_j = \emptyset$ for all $i, j = 1, \ldots, n$ and $i \neq j$,
- within each Ω_i, the intensity values are varying smoothly and/or slowly for $i = 1, \ldots, n$,
- the intensity values are varying rapidly for the adjacent subregions Ω_i and Ω_j for all $i, j = 1, \ldots, n$ and $i \neq j$.

Edge Detectors

We call pixels *edge pixels* if the intensity of an image function changes suddenly. When a method detects edge pixels it is called an *edge detector*. *Edges*, defined as sets of edge pixels, can be attained by using first- and second-order derivatives. Some edge detectors, which are using the first-order derivatives to obtain the edges, are for instance the *gradient operators*, the *Roberts operators*, and the *Sobel operators*. The two-dimensional Gaussian function, with the standard deviation σ,

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{2.1}$$

can also be used to detect edges. This is for instance done in the *Marr-Hildreth edge detector* and the *Canny edge detector* which uses the Laplacian and the first derivative of the Gaussian function, respectively. For more details concerning the mentioned edge detectors, see [1].

The edge detector used in further chapters needs to be close to zero at the edges and larger elsewhere. Therefore, we use an experimental edge detector given by

$$g\left(|\nabla f(x,y)|\right) = \frac{a}{1+b\,|\nabla\left[G_\sigma(x,y) * f(x,y)\right]|^2}, \tag{2.2}$$

where the constants $a,\ b > 0$, the gradient operator and the convolution operator is given by ∇ and $*$, respectively, and $G_\sigma * f$ is the smoothed version of the image function f. Fig. 2.1 shows the edge detector (2.2) with different values of σ, a and b.

(a) The original image

(b) $\sigma = 0.5$, $a = 0.5$, $b = 1$.

(c) $\sigma = 0.5$, $a = 1$, $b = 0.2$.

(d) $\sigma = 0.5$, $a = 1$, $b = 1$.

(e) $\sigma = 0.5$, $a = 1$, $b = 50$.

(f) $\sigma = 2,\ a = 0.5,\ b = 1.$ **(g)** $\sigma = 2, a = 1, b = 0.2.$ **(h)** $\sigma = 2, a = 1, b = 1.$ **(i)** $\sigma = 2, a = 1, b = 50.$

(j) $\sigma = 5,\ a = 0.5,\ b = 1.$ **(k)** $\sigma = 5, a = 1, b = 0.2.$ **(l)** $\sigma = 5, a = 1, b = 1.$ **(m)** $\sigma = 5,\ a = 1,\ b = 50.$

Figure 2.1: Using the edge detector (2.2) with different values of σ, a and b.

Level Set Method

In [6], Osher and Sethain proposed a simple and adaptable method, known as the level set method, to compute and analyse the motion of a curve C in the open and bounded subset $\Omega \subset \mathbb{R}^2$. The curve C bounds the region $\omega \subset \Omega$, such that $C = \partial\omega$. Now, a velocity field $\vec{v}$ is used to find the motion of C, which may depend on time, position, the external physics and the geometry of the curve. The idea in [6] is to define a Lipschitz continuous function $\phi(t, x, y)$ to represent implicitly the curve C as the set where $\phi(t, x, y) = 0$, i.e.

$$C\,(t) = \big\{(x, y) : \phi(t, x, y) = 0\big\},$$

where $\phi(t, x, y) > 0$ inside ω and $\phi(t, x, y) < 0$ inside $\Omega \setminus \overline{\omega}$. By using the level set function ϕ, topological changes are allowed.

The motion of the level set function ϕ is given by evolving the zero level set of ϕ using the desired velocity $\vec{v}$ on C. Hence, the level set equation is obtained, see for instance [27]:

$$\phi_t + \vec{v} \cdot \nabla\phi = 0.$$

The velocity field is written as the sum of the normal vector $\vec{N}$ and tangent vector $\vec{T}$, such that $\vec{v} = v_N\,\vec{N} + v_T\,\vec{T}$ with their respective velocity components. We are merely interested in moving in the normal direction. As a consequence, $\vec{T} \cdot \nabla\phi = 0$, and the level set equation is rewritten as

$$\phi_t + v_N\,\vec{N} \cdot \nabla\phi = 0.$$

Since the unit normal vector $\vec{N} = \dfrac{\nabla\phi}{|\nabla\phi|}$, we obtain the equality

$$\begin{aligned}\vec{N}\cdot\nabla\phi &= \frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\phi = \frac{|\nabla\phi|^2}{|\nabla\phi|}\\ &= |\nabla\phi|.\end{aligned}$$

Hence,

$$\phi_t + v_N\,|\nabla\phi| = 0.$$

For the motion by mean curvature κ, which is an outward normal vector, the normal velocity v_N is characterized by $v_N = -c\,\kappa$ with a constant $c \in \mathbb{R}$ and $\kappa = \nabla\cdot\left(\dfrac{\nabla\phi}{|\nabla\phi|}\right)$.

2.2 Selected Optimization Methods

In this section we describe some consepts of convexity and look at some optimization methods, such as gradient descent method and the Lagrangian method.

2.2.1 Convexity

Convex Set

Let $S \subset \mathbb{R}^n$ be a set, and x and y be two points in S. A line segment of x and y is denoted by

$$[\![x, y]\!] = \big\{\alpha\,x + (1-\alpha)\,y \,\big|\, 0 \le \alpha \le 1\big\}.$$

The set S is a *convex set* if for any pair of points (x, y) of S the line segment $[\![x, y]\!]$ is contained in S.

Convex Function

The real-valued function $f : S \to \mathbb{R}$ is said to be a *convex function* if its domain $S \subset \mathbb{R}^n$ is a convex set and if, for any two points $x, y \in S$, the function satisfies the following property:

$$f(\alpha\,x + (1-\alpha)\,y) \le \alpha\,f(x) + (1-\alpha)\,f(y),$$

for any $0 \le \alpha \le 1$.

Global & Local Minimum

Let $f : S \to \mathbb{R}$ be a real-valued function defined on a set $S \subset \mathbb{R}^n$.

Global minimum

A point $x^* \in S$ is called a *global minimum* of the function f if $f(x^*) \leq f(x)$ for all points $x \in S$.

Local minimum

A point $x^* \in S$ is called a *local minimum* of the function f if there exists a neighbourhood $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all points $x \in \mathcal{N}$.

In addition, any local minimum x^* of a convex function f is a global minimum of f.

First-Order Necessary Condition [11]

If x^* is a local minimizer and f is continuously differentiable in an open neighbourhood of x^*, then $\nabla f(x^*) = 0$.

Convex optimization problem

Let $f : S \to \mathbb{R}$ be a real-valued convex function over a given convex set $S \subset \mathbb{R}^n$. Finding the global minimum $x^* \in S$ of $f(x)$, for all points $x \in S$, is called a *convex optimization problem*, i.e.

$$f(x^*) = \min_{x \in S} f(x).$$

2.2.2 Gradient Descent Method

A local minimum point x^* of a function $f : \mathbb{R}^n \to \mathbb{R}$ can be found by using an algorithm for unconstrained optimization. This algorithm requires a starting point x_0, which lies at the best close to the minimum point. Starting with x_0, the algorithm creates an iteration sequence $\{x_k\}_{k=0}^{\infty}$, and it terminates when there is no more difference between x_k and x_{k+1}, or when $x_{k+1} \approx x^*$. The next iteration x_{k+1} is decided by using information about the function f at x_k. Such an iteration is defined by

$$x_{k+1} = x_k + \gamma_k p_k,$$

for a chosen step size $\gamma_k > 0$ and a chosen direction p_k. The step size γ_k can either be a constant for all k or a different scalar at each iteration. If we choose $p_k = -\nabla f(x_k)$, the algorithm is called the *gradient descent method*, also known as the *steepest descent method*. The advantage of this method is

that it uses additional information about the function f, explicitly ∇f , but it can be extremely slow on difficult problems. For more details about the gradient descent method, see [23].

2.2.3 The Lagrangian Method

Let us now consider a constrained optimization problem of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \quad \text{for } i = 1, \ldots, m, \end{aligned} \tag{2.3}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the constraints $h_i : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable functions. By introducing *Lagrange multipliers* λ_i, $i = 1, \ldots, m$, we can define a *Lagrangian function*

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i h_i(x).$$

If x^* is the local minimum of the problem (2.3) and if the constraint gradients $\nabla h_1(x^*), \ldots, \nabla h_m(x^*)$ are linearly independent, then there exists a Lagrange multiplier vector $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$ such that the following conditions are satisfied for the point (x^*, λ^*), see for instance [23, 11]:

$$\begin{aligned} \nabla \mathcal{L}(x^*, \lambda^*) &= 0, \\ h_i(x^*) &= 0, \quad \text{for all } i = 1, \ldots, m. \end{aligned}$$

The Augmented Lagrangian Method

The constrained optimization problem (2.3) can be replaced by optimization problem only consisting of the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and a penalizing term. The penalizing term uses each constraint and is positive when violating the constraints and zero otherwise. Multiplying the penalty term with a large positive coefficient, the violations of the constraints become more severe and force the minimizer of the penalty function closer to the region of feasibility. The problem is now given as

$$\min_x \left\{ f(x) + \frac{c}{2} \sum_{i=1}^{m} h_i^2(x) \right\}, \tag{2.4}$$

where the *penalty parameter* $c > 0$ is penalizing the constraint violations severely when forcing $c \to \infty$.

Since (2.4) does not quite satisfy the constraints $h_i(x) = 0$ for $i = 1, \ldots, m$, the *augmented Lagrangian function* $\mathcal{L}_c(x, \lambda)$ is an improvement achieved by including an estimate of the Lagrange multiplier vector λ :

$$\mathcal{L}_c(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i h_i(x) + \frac{c}{2} \sum_{i=1}^{m} h_i^2(x).$$

The augmented Lagrangian function is now used to improve the estimate of λ such that it will be closer to the optimal Lagrangian multipliers λ^*. Let us first fix λ at the current estimate λ^k and let c be fixed. By minimizing $\mathcal{L}_c$ with respect to x, the *first-order necessary condition* is used for x_k being the approximate minimizer:

$$\nabla \mathcal{L}_c(x_k, \lambda^k) = \nabla f(x_k) - \sum_{i=1}^{m} \left[\lambda_i^k - c\, h_i(x_k)\right] \nabla h_i(x_k) = 0.$$

Comparing this with the optimality conditions for the Lagrangian method,

$$\lambda_i^* \approx \lambda_i^k - c\, h_i(x_k)$$

gives us a formula to update the current estimate λ^k of the Lagrange multiplier vector:

$$\lambda_i^{k+1} = \lambda_i^k - c\, h_i(x_k). \tag{2.5}$$

This procedure, when first minimizing $\mathcal{L}_c(x, \lambda)$ w.r.t. x and then updating the estimate of λ using (2.5), is called the *method of multipliers* or the *augmented Lagrangian method.*

2.3 Total Variation

Let $\Omega \subseteq \mathbb{R}^n$ be a bounded open domain. In [20, 21], the total variation of a function $f \in L^1(\Omega)$ is defined as

$$TV(f, \Omega) = \sup_{\xi} \left\{ \int_\Omega f\, \nabla \cdot \xi \, dx \;\middle|\; \xi \in C_c^1(\Omega, \mathbb{R}^n)\,,\ |\xi(x)| \leq 1\,,\ x \in \Omega \right\},$$

where $\nabla \cdot \xi = \sum_{i=1}^{n} \frac{\partial \xi}{\partial x_i}$ is the divergence operator and the space $C_c^1(\Omega, \mathbb{R}^n)$ containes all continuously differentiable functions with compact support. The function f is said to be of bounded variation if $TV(f, \Omega) < \infty$.

If the function f is continuously differentiable on Ω, we can apply the Divergence theorem, the compact support condition, Cauchy-Schwarz inequality, and $|\xi(x)| \leq 1$ to see that the integral term in $TV(f, \Omega)$ is bounded:

$$\begin{aligned}\int_\Omega f \, \nabla \cdot \xi \, dx &= -\int_\Omega \nabla f \cdot \xi dx + \int_{\partial\Omega} f \, \xi \cdot \vec{n} \, ds \\ &\leq \left| \int_\Omega \nabla f \cdot \xi dx \right| \\ &\leq \int_\Omega |\nabla f| \, |\xi| \, dx \\ &\leq \int_\Omega |\nabla f| \, dx,\end{aligned}$$

where $\vec{n}$ is the outward normal to the boundary of Ω. Therefore, when $\xi \to -\dfrac{\nabla f}{|\nabla f|}$, we receive

$$TV(f, \Omega) = \int_\Omega |\nabla f| \, dx.$$

Let $u(x)$ be the characteristic function of $\omega \subset \Omega$, such that

$$u(x) = \begin{cases} 1, & \text{if} \quad x \in \overline{\omega} \\ 0, & \text{if} \quad x \in \Omega \setminus \overline{\omega}. \end{cases}$$

By the total variation of $u(x)$, the Divergence theorem and Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}TV(u, \Omega) &= \sup_{|\xi(x)| \leq 1} \int_\Omega u \, \nabla \cdot \xi \, dx = \sup_{|\xi(x)| \leq 1} \int_{\overline{\omega}} \nabla \cdot \xi \, dx \\ &= \sup_{|\xi(x)| \leq 1} \int_{\partial\omega} \xi \cdot \vec{n} \, ds \leq \sup_{|\xi(x)| \leq 1} \int_{\partial\omega} |\xi||\vec{n}| \, ds \\ &\leq |\partial\omega|,\end{aligned}$$

where $\partial\omega$ is the boundary of ω with the outward normal $\vec{n}$, and the length of the boundary is given by $|\partial\omega|$. Choosing $\xi = \vec{n}$ on $\partial\omega$, the total variation of u over the domain Ω is equal to $|\partial\omega|$:

$$TV(u, \Omega) = |\partial\omega|.$$

Chapter 3

An Active Contour Method for Image Segmentation

3.1 The Basic Idea of the Model

Assume u_0 is a given image. Objects in the image can be detected by an evolving curve, subject to contraints from the image. The curve will move towards the interior normal of the curve and stops at the boundary of the object. This idea is called an active contour model, also known as a snake model. Classical active contour models are presented in [12], [13] and [14]. These models are depending on the gradient of the image u_0 to stop the evolving curve. The model suggested by Chan and Vese [2] is independent of the edges of the input image, and this chapter is based on this Chan-Vese model.

Let the given image be $u_0 : \Omega \to \mathbb{R}$, where Ω is a bounded subset of $\mathbb{R}^2$ with $\partial\Omega$ as its boundary, and let the parameterized curve be $\Lambda(s) : [0,1] \to \mathbb{R}^2$.

The classical active contour model in [12] and [13] suggests minimizing the following functional:

$$\begin{aligned} E_1^{\text{snake}}(\Lambda) &= \alpha \int_0^1 |\Lambda'(s)|^2 \, ds + \beta \int_0^1 |\Lambda''(s)|^2 \, ds \\ &\quad + \gamma \int_0^1 g\big(\big| \nabla u_0 \left(\Lambda(s)\right) \big|\big) \, ds, \end{aligned} \tag{3.1}$$

with the parameters α, β, $\gamma > 0$ and g is an edge detector function depending on the gradient of the image u_0. The internal energy, given by the first two terms, is a smoothness constraint, as the first term represents the length of the parameterized curve and the second term controls the rigidity of the

moving curve. The external energy, given by the third term, attracts the contour towards features of the image, e.g. lines and edges. The edge detector $g(|\nabla u_0|)$ is a positive and decreasing function and depends on the image gradient of u_0. For instance, the edge detector given in (2.2) can be used. So, the function $g(|\nabla u_0|)$ is zero at the edges and positive in homogeneous regions.

In order to represent the curve evolution, the level set method and the motion of mean curvature by Osher and Sethian [6] is used. This method allows sharp corners and topological changes. Let the curve Λ be given implicitly by the Lipschitz continuous function $\phi : \Omega \to \mathbb{R}$:

$$\Lambda = \{(x, y) : \phi(t, x, y) = 0\}.$$

The function $\phi(t, x, y)$ represents the evolving curve as the zero level curve at time t. Moving the curve in the normal direction, where the speed is the motion by mean curvature κ, we obtain the following differential equation:

$$\begin{cases} \phi_t = |\nabla\phi|\,\kappa, & t \in (0,\infty),\ (x,y) \in \mathbb{R}^2, \\ \phi(0,x,y) = \phi_0(x,y), & (x,y) \in \mathbb{R}^2, \end{cases}$$

where $\kappa = \nabla \cdot \left(\dfrac{\nabla\phi}{|\nabla\phi|}\right)$ and $\phi_0(x, y)$ is the initial contour.

If the curve moves in the normal direction with the speed $g(|\nabla u_0|)(\kappa + \zeta)$ for $\zeta \geq 0$, a geometric active contour model [13], which is based on the motion by κ, can be given by:

$$\begin{cases} \phi_t = g\,|\nabla\phi|\,(\kappa + \zeta), & t \in (0,\infty),\ (x,y) \in \mathbb{R}^2, \\ \phi(0,x,y) = \phi_0(x,y), & (x,y) \in \mathbb{R}^2, \end{cases}$$

where g is the edge function in (2.2) and ζ is a nonnegative constant. The constant ζ, thought of as a correction term, is chosen such that $\kappa + \zeta$ remains positive. When κ becomes null or negative, ζ pushes the curve to the boundary of the object.

In [15], two other active contour models are proposed and these models also use the image gradient $|\nabla G_\sigma * u_0|$ to stop the evolving curve. On the other hand, [14] proposes a geodesic active contour model which uses (2.2) to obtain a minimizer Λ.

These active contour models are all depending on the image gradient to stop the evolution of the curve. Therefore, only objects with edges defined by a gradient can be detected. When the Gaussian is used for very noisy images, the edges will be smoothed, which is a drawback for these models.

The paper [2] by Chan and Vese proposes a different active contour model not depending on an edge function to stop the curve evolution. Instead, the proposed model uses Mumford-Shan segmentation to detect contours.

3.2 The Model's Description

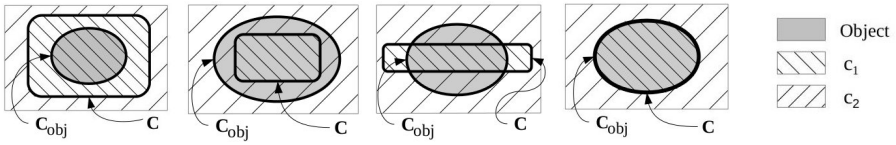
In [2] Chan and Vese proposes a new model to detect objects in a given image using active contours. This method minimizes an energy functional in order to segment images.

Let Ω be a bounded subset in $\mathbb{R}^2$ where an evolving closed curve C in Ω can be defined. This curve is the boundary of the subset $\omega \subset \Omega$ such that $C = \partial\omega$. Now we have a region inside C, ω, and a region outside C, $\Omega \setminus \overline{\omega}$.

We now look at a simple case of an energy based segmentation. Assume that the input image $u_0 : \Omega \to \mathbb{R}$ is divided into two nonoverlapping regions where each region has nearly piecewise-constant intensities. These two regions are the wanted object and the background, and the curve C_{obj} is the boundary of the object. Inside C_{obj} the intensity is denoted as u_0^{in} and outside C_{obj} the intensity is denoted as u_0^{out}. The energy functional to be minimized in this model contains the "fitting" term:

$$\begin{aligned} E_{in}(C) + E_{out}(C) = & \int\limits_{\text{inside}(C)} |u_0(x,y) - c_1|^2 dxdy \\ & + \int\limits_{\text{outside}(C)} |u_0(x,y) - c_2|^2 dxdy, \end{aligned} \tag{3.2}$$

where C is any given variable curve, and the constant c_1 is the average intensity value of u_0 inside C and the constant c_2 is the average intensity value of u_0 outside C. In fig. 3.1 some different positions of C are shown. Fig. 3.1a shows the case when $c_1 \not\approx u_0^{in}$ and $c_2 \approx u_0^{out}$ which means that $E_{in}(C) > 0$ and $E_{out}(C) \approx 0$. When C lies inside the object, as shown in fig. 3.1b, then $c_1 \approx u_0^{in}$ and $c_2 \not\approx u_0^{out}$, resulting in $E_{in}(C) \approx 0$ and $E_{out}(C) > 0$. In fig. 3.1c, C is both inside and outside the object, such that $c_1 \not\approx u_0^{in}$ and $c_2 \not\approx u_0^{out}$, and therefore $E_{in}(C) > 0$ and $E_{out}(C) > 0$. The last case, fig. 3.1d, shows that C lies at the boundary of the object such that $c_1 \approx u_0^{in}$ and $c_2 \approx u_0^{out}$, and consequently $E_{in}(C) \approx 0$ and $E_{out}(C) \approx 0$. This simple case shows that C must be equal to C_{obj} to minimize the functional.



(a) The curve C is outside the object. **(b)** The curve C is inside the object. **(c)** The curve C is both inside and outside the object. **(d)** The curve C is on the boundary of the object.

Figure 3.1: All possible cases for the position of the curve.

The energy functional to be minimized has in addition two regularization terms: the length of the curve C and the area inside C. The reason for using these terms is to penalize the length of C, such that the segmentation does not become too noisy, and to control the area of C. The energy functional to be minimized is now introduced as

$$\begin{aligned} E\left(c_1, c_2, C\right) =& \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}\left(\text{inside}\left(C\right)\right) \\ &+ \lambda_1 \cdot \int\limits_{\text{inside}(C)} |u_0(x,y) - c_1|^2 dxdy \\ &+ \lambda_2 \cdot \int\limits_{\text{outside}(C)} |u_0(x,y) - c_2|^2 dxdy, \end{aligned} \tag{3.3}$$

where μ, $\nu \geq 0$ and λ_1, $\lambda_2 > 0$ are fixed parameters which control the influence of each term.

We now want to consider the energy minimization problem

$$\inf_{c_1, c_2, C} E\left(c_1, c_2, C\right).$$

Relating the Model with the Mumford-Shah Functional

Before solving the energy problem, we will look at the Mumford-Shah functional from [5] for segmentation:

$$\begin{aligned} E^{MS}(u, C) =& \mu \cdot \text{Length}\left(C\right) + \int_{\Omega \setminus C} |\nabla u\left(x,y\right)|^2 \, dxdy \\ &+ \lambda \cdot \int_{\Omega} |u_0\left(x,y\right) - u\left(x,y\right)|^2 \, dxdy, \end{aligned} \tag{3.4}$$

where $u_0 : \Omega \to \mathbb{R}$ is the image to be segmented, μ and λ are fixed positive parameters, u is the solution image achieved by minimizing the above functional containing fixed number of smooth regions Ω_i, and C is the interface between these regions.

A simpler version of the functional above, called the minimal partition problem, is to restrain u to be a piecewise constant function such that u is equal to a constant c_i inside the region Ω_i. As a result, the second term in (3.4) disappears. In [5], c_i is the average intensity value of u_0 over inside the boundary of Ω_i.

The Chan-Vese model (3.3) is exactly a minimal partition problem if $\nu = 0$ and $\lambda_1 = \lambda_2 = \lambda$. So, we want the function u to be the best approximation

of u_0 where u only takes two values and C is the contour:

$$u(x,y) = \begin{cases} c_1 = \text{average}\,(u_0) \text{ inside of } C, \\ c_2 = \text{average}\,(u_0) \text{ outside of } C. \end{cases} \tag{3.5}$$

The next step is to use the level set method to formulate and solve this specific problem.

3.2.1 Level Set Formulation of the Model

The level set method in [6] assumes that the zero level set of a Lipschitz continuous function $\phi : \Omega \to \mathbb{R}$ is an implicit and parameter-free representation of the curve $C \subset \Omega$. As it is given in [7], where $\omega \subset \Omega$ is open, the implicit function $\phi(x,y)$ satisfies

$$\begin{cases} \phi(x,y) = 0, & \text{for } (x,y) \in \partial\omega = C, \\ \phi(x,y) > 0, & \text{for } (x,y) \in \omega = \text{inside}(C), \\ \phi(x,y) < 0, & \text{for } (x,y) \in \Omega \setminus \overline{\omega} = \text{outside}(C). \end{cases}$$

The above assumptions are illustrated in fig. 3.2.

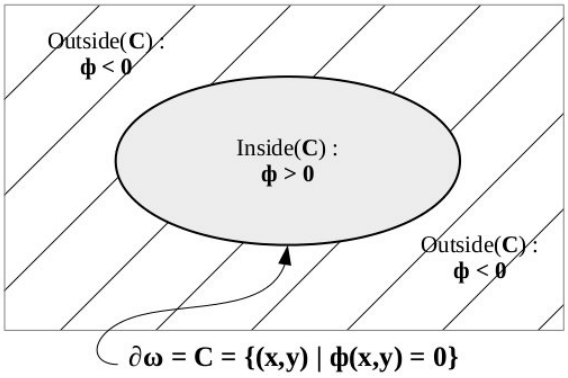


Figure 3.2: An illustration of the level set function ϕ.

Now, it is more suitable to introduce a characteristic function of the set $\overline{\omega}$, which is given by the Heaviside function only depending on ϕ:

$$H(\phi) = \begin{cases} 1, & \text{if } \phi \geq 0, \\ 0, & \text{if } \phi < 0. \end{cases} \tag{3.6}$$

By differentiating the Heaviside function, we obtain the one-dimensional Dirac's delta function:

$$\frac{d}{d\phi} H(\phi) = \delta(\phi) = \begin{cases} 1, & \text{for } \phi = 0, \\ 0, & \text{for } \phi \neq 0. \end{cases} \tag{3.7}$$

Using both (3.6) and (3.7), we can express the length of the boundary $\partial\omega$ and the area of ω in the energy functional (3.3):

$$\begin{aligned}\text{Length}\,(\partial\omega) &= \int_{\Omega} |\nabla H\,(\phi\,(x,y))\,|\,dxdy \\ &= \int_{\Omega} \delta\,(\phi\,(x,y))\;|\nabla\phi\,(x,y)\,|\,dxdy,\end{aligned}$$

and

$$\text{Area}\,(\omega) \;=\; \int_{\Omega} H\,(\phi\,(x,y))\;dxdy.$$

In this thesis, a weighted length term is used:

$$\text{Length}_g\,(\partial\omega) \;=\; \int_{\Omega} g(|\nabla u_0(x,y)|)\,\delta\,(\phi\,(x,y))\;|\nabla\phi\,(x,y)\,|\,dxdy,$$

where $g(|\nabla u_0|)$ is the edge detector (2.2).

The Heaviside function is also used to extract the regions inside and outside the boundary. Therefore the other terms in (3.3) can be written as:

$$\int\limits_{\text{inside}(C)} |u_0\,(x,y) - c_1|^2\,dxdy \;=\; \int_{\Omega} |u_0(x,y) - c_1|^2\,H\,(\phi\,(x,y))\;dxdy$$

and

$$\int\limits_{\text{outside}(C)} |u_0\,(x,y) - c_2|^2\,dxdy \;=\; \int_{\Omega} |u_0(x,y) - c_2|^2\,(1 - H\,(\phi\,(x,y)))\;dxdy.$$

By using all the terms above together, the energy $E(c_1, c_2, \phi)$ is now expressed as:

$$\begin{aligned}E(c_1,c_2,\phi) \;=& \mu \int_{\Omega} g(|\nabla u_0(x,y)|)\,\delta(\phi(x,y))\,|\nabla\phi(x,y)|\,dxdy \\ &+\; \nu \int_{\Omega} H(\phi(x,y))\,dxdy \\ &+\; \lambda_1 \int_{\Omega} |u_0(x,y) - c_1|^2 H(\phi(x,y))\,dxdy \\ &+\; \lambda_2 \int_{\Omega} |u_0(x,y) - c_2|^2 (1 - H(\phi(x,y)))\,dxdy.\end{aligned}$$

Since this particular model is a Mumford-Shah minimal partition problem, we can use the level set formulation and (3.5) to obtain the solution of this model:

$$u\,(x,y) \;=\; c_1\,H\big(\phi\,(x,y)\,\big) + c_2\,\big(1 - H\big(\phi\,(x,y)\,\big)\big)\,,$$

for $(x,y) \in \Omega$.

The existence of the solution of this specific problem has been proven in [5] with the assumption that u_0 is continuous on Ω. Therefore, we expect that there exist minimizers of the energy functional $E(c_1, c_2, \phi)$.

The Heaviside and the delta functions are not differentiable at $\phi = 0$ which is necessary in further computations. Hence, we consider $H_\varepsilon \in C^2(\Omega)$ and $\delta_\varepsilon \in C^1(\Omega)$, $\varepsilon \to 0$, roughly as the regularized versions of H and δ. An example of $H_\varepsilon \in C^2(\Omega)$ and $\delta_\varepsilon \in C^1(\Omega)$ can be seen in section 3.3. Theoretically, this is verified by replacing both H and δ by the smooth functions H_ε and δ_ε and passing them to the limit [10]. The related regularized version of the energy functional is given by

$$\begin{aligned} E_\varepsilon(c_1, c_2, \phi) \;=& \mu \int_\Omega g(|\nabla u_0(x,y)|)\,\delta_\varepsilon(\phi(x,y))\,|\nabla\phi(x,y)|\,dxdy \\ &+\; \nu \int_\Omega H_\varepsilon(\phi(x,y))\,dxdy \\ &+\; \lambda_1 \int_\Omega |u_0(x,y) - c_1|^2 H_\varepsilon(\phi(x,y))\,dxdy \\ &+\; \lambda_2 \int_\Omega |u_0(x,y) - c_2|^2 (1 - H_\varepsilon(\phi(x,y)))\,dxdy. \end{aligned}$$

Minimizing E_ε w.r.t. c_1 and c_2

Assuming that ω and $\Omega \setminus \overline{\omega}$ are nonempty, the constants c_1 and c_2 can be expressed by minimizing the energy $E_\varepsilon(c_1, c_2, \phi)$ with respect to c_1 and c_2 one by one while keeping ϕ fixed:

$$c_1 \;=\; \frac{\int_\Omega u_0(x,y)\,H_\varepsilon(\phi(x,y))\,dxdy}{\int_\Omega H_\varepsilon(\phi(x,y))\,dxdy}$$

and

$$c_2 \;=\; \frac{\int_\Omega u_0(x,y)\,(1 - H_\varepsilon(\phi(x,y)))\,dxdy}{\int_\Omega (1 - H_\varepsilon(\phi(x,y)))\,dxdy}.$$

The constants c_1 and c_2 are also the average intensities inside and outside the curve $\partial\omega$, respectively.

If ω or $\Omega \setminus \overline{\omega}$ is empty, the "degenerate" cases for c_1 or c_2, respectively, the constants c_1 and c_2 are assigned to be

$$\begin{cases} c_1 = \text{average}(u_0) & \text{inside} \quad \omega, \\ c_2 = \text{average}(u_0) & \text{inside} \quad \Omega \setminus \overline{\omega}. \end{cases}$$

Minimizing E_ε w.r.t. ϕ

Now, we minimize E_ε with respect to ϕ while keeping c_1 and c_2 fixed. For the minimization, we are applying calculus of variations. For more details concerning calculus of variations, see for instance [9].

Let us consider the Gâteaux differential [26, p. 23] of the functional $E_\varepsilon(\phi)$ with respect to ϕ in the direction ψ:

$$\begin{aligned} D_\psi E_\varepsilon(\phi) \;&=\; \lim_{\tau\to 0} \frac{1}{\tau}\Big[E_\varepsilon(\phi+\tau\psi) - E_\varepsilon(\phi)\Big] \\ &= \mu \lim_{\tau\to 0} \frac{1}{\tau} \int_\Omega g\Big[\delta_\varepsilon(\phi+\tau\psi)|\nabla(\phi+\tau\psi)| \;-\; \delta_\varepsilon(\phi)|\nabla\phi|\Big]dxdy \\ &\quad +\; \nu \lim_{\tau\to 0} \frac{1}{\tau} \int_\Omega \Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy \\ &\quad +\; \lambda_1 \lim_{\tau\to 0} \frac{1}{\tau} \int_\Omega |u_0(x,y)-c_1|^2\Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy \\ &\quad -\; \lambda_2 \lim_{\tau\to 0} \frac{1}{\tau} \int_\Omega |u_0(x,y)-c_2|^2\Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy. \end{aligned}$$

To find the limit of the first integral, L'Hopital's rule is applied:

$$\begin{aligned} &\lim_{\tau\to 0} \frac{1}{\tau} \int_\Omega g\Big[\delta_\varepsilon(\phi+\tau\psi)|\nabla(\phi+\tau\psi)| \;-\; \delta_\varepsilon(\phi)|\nabla\phi|\Big]dxdy \\ &\qquad = \lim_{\tau\to 0} \int_\Omega g\Big[\delta'_\varepsilon(\phi+\tau\psi)\,\psi\,|\nabla(\phi+\tau\psi)| \\ &\qquad\qquad\qquad +\; \delta_\varepsilon(\phi+\tau\psi)\frac{\nabla(\phi+\tau\psi)\cdot\nabla\psi}{|\nabla(\phi+\tau\psi)|}\Big]dxdy \\ &\qquad = \int_\Omega g\Big[\delta'_\varepsilon(\phi)\,\psi\,|\nabla\phi| \;+\; \delta_\varepsilon(\phi)\frac{\nabla\phi\cdot\nabla\psi}{|\nabla\phi|}\Big]dxdy. \end{aligned}$$

Now, we can apply the Divergence theorem to the second part of this integral:

$$\begin{aligned} \int_\Omega g\,\delta_\varepsilon(\phi)\frac{\nabla\phi\cdot\nabla\psi}{|\nabla\phi|}dxdy \;&=\; -\int_\Omega \nabla\cdot\left(g\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right)\psi\;dxdy \\ &\quad +\; \int_{\partial\Omega}\left(g\,\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\vec{n}\right)\psi\;ds, \end{aligned}$$

where $\vec{n}$ is the outward normal vector to the boundary $\partial\Omega$. Using that $\nabla\phi \cdot \nabla\phi = |\nabla\phi|^2$ we can simplify the first term:

$$\begin{aligned}
&\nabla \cdot \left(g\,\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \\
&\quad = \nabla g \cdot \left(\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\right) \;+\; g\,\delta'_\varepsilon(\phi)\,\nabla\phi \cdot \frac{\nabla\phi}{|\nabla\phi|} \;+\; g\,\delta_\varepsilon(\phi)\,\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \\
&\quad = \delta_\varepsilon(\phi)\frac{\nabla g\cdot\nabla\phi}{|\nabla\phi|} \;+\; g\,\delta'_\varepsilon(\phi)\,|\nabla\phi| \;+\; g\,\delta_\varepsilon(\phi)\,\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right).
\end{aligned}$$

The limit of the first integral is now:

$$\begin{aligned}
&\int_\Omega g\left[\delta'_\varepsilon(\phi)\,\psi\,|\nabla\phi| \;+\; \delta_\varepsilon(\phi)\,\frac{\nabla\phi\cdot\nabla\psi}{|\nabla\phi|}\right] dxdy \\
&\qquad = \;-\;\int_\Omega \delta_\varepsilon(\phi)\left(\frac{\nabla g\cdot\nabla\phi}{|\nabla\phi|} \;+\; g\,\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right)\psi\,dxdy \\
&\qquad\quad + \int_{\partial\Omega}\left(g\,\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\vec{n}\right)\psi\,ds.
\end{aligned}$$

L'Hopital is also applied for the next three integrals where we use that $\frac{d}{d\phi}H_\varepsilon(\phi) = \delta_\varepsilon(\phi)$:

$$\begin{aligned}
\lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega \Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy \;&=\; \lim_{\tau\to 0}\int_\Omega \delta_\varepsilon(\phi+\tau\psi)\,\psi\,dxdy \\
&= \int_\Omega \delta_\varepsilon(\phi)\,\psi\,dxdy,
\end{aligned}$$

$$\begin{aligned}
&\lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega |u_0(x,y)-c_1|^2\Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy \\
&\qquad = \;\lim_{\tau\to 0}\int_\Omega |u_0(x,y)-c_1|^2\delta_\varepsilon(\phi+\tau\psi)\,\psi\,dxdy \\
&\qquad = \;\int_\Omega |u_0(x,y)-c_1|^2\delta_\varepsilon(\phi)\,\psi\,dxdy,
\end{aligned}$$

and

$$\begin{aligned}
&\lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega |u_0(x,y)-c_2|^2\Big[H_\varepsilon(\phi+\tau\psi) \;-\; H_\varepsilon(\phi)\Big]dxdy \\
&\qquad = \;\lim_{\tau\to 0}\int_\Omega |u_0(x,y)-c_2|^2\delta_\varepsilon(\phi+\tau\psi)\,\psi\,dxdy \\
&\qquad = \;\int_\Omega |u_0(x,y)-c_2|^2\delta_\varepsilon(\phi)\,\psi\,dxdy.
\end{aligned}$$

Summerazing all above, we obtain:

$$\begin{aligned} D_\psi E_\varepsilon(\phi) \;&=\; \lim_{\tau\to 0}\frac{1}{\tau}\Big[E_\varepsilon(\phi+\tau\psi)-E_\varepsilon(\phi)\Big] \\ &=\; -\int_\Omega \delta_\varepsilon(\phi)\Bigg(\mu\,\frac{\nabla g\cdot\nabla\phi}{|\nabla\phi|} \;+\; \mu\, g\,\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \;-\; \nu \\ &\qquad\qquad -\; \lambda_1\,|u_0(x,y)-c_1|^2 \;+\; \lambda_2\,|u_0(x,y)-c_2|^2\Bigg)\,\psi\,dxdy \\ &\quad +\int_{\partial\Omega}\mu\left(g\,\delta_\varepsilon(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\vec{n}\right)\psi\;ds. \end{aligned}$$

By the first order necessary condition for a local minimizer of a functional [11], we need that:

$$D_\psi E_\varepsilon(\phi) \;=\; 0.$$

By associating the energy functional $E_\varepsilon(c_1,c_2,\phi)$ with the Euler-Lagrange equation [9], we can parameterize the descent direction by an artificial time parameter, $t>0$, and seek for a steady-state solution of this problem:

$$\begin{cases} \dfrac{\partial\phi}{\partial t} \;=\; \delta_\varepsilon(\phi)\Bigg(\mu\,\dfrac{\nabla g\cdot\nabla\phi}{|\nabla\phi|} \;+\; \mu\, g\,\nabla\cdot\left(\dfrac{\nabla\phi}{|\nabla\phi|}\right) \;-\; \nu \\ \qquad\qquad -\;\lambda_1\,|u_0(x,y)-c_1|^2 \;+\; \lambda_2\,|u_0(x,y)-c_2|^2\Bigg) & \text{in} \quad (0,\infty)\times\Omega, \\ \phi(0,x,y) \;=\; \phi_0(x,y) & \text{in} \quad \Omega, \\ g\,\delta_\varepsilon(\phi)\dfrac{\nabla\phi}{|\nabla\phi|}\cdot\vec{n} \;=\; 0 & \text{on} \quad \partial\Omega, \end{cases}$$

where $\phi(0,x,y) \;=\; \phi_0(x,y)$ is the initial curve, $\vec{n}$ is the outward normal to the boundary $\partial\Omega$, and $\nabla\phi\cdot\vec{n}$ denotes the normal derivative of ϕ at the boundary.

3.3 A Numerical Approximation of the Model

A regularized version of $H(\phi)\in C^\infty(\,\Omega\,)$ is suggested in [2]:

$$H_\varepsilon(\phi)=\frac{1}{2}\left[1+\frac{2}{\pi}\tan^{-1}\left(\frac{\phi}{\varepsilon}\right)\right].$$

By differentiating $H_\varepsilon(\phi)$ the regularized delta function $\delta_\varepsilon(\phi)$ is obtained:

$$\begin{aligned} \delta_\varepsilon(\phi) \;&=\frac{d}{d\phi}\,H_\varepsilon(\phi) \\ &=\frac{\varepsilon}{\pi\,(\,\varepsilon^2+\phi^2\,)}, \end{aligned}$$

for a constant $\varepsilon > 0$. If ε is chosen to be too small, the δ_ε-function will create large oscillations. But if ε is chosen to be too large, the accuracy of the method will be reduced. This is prevented by substituting the δ_ε-function by 1 in the implementation [10]. The energy functional E_ε is non-convex because of the Heaviside function, and therefore allowing many local minima.

The variable ϕ in the Euler-Lagrange equation is discretized by using backward, forward and central differences. Knowing that the input image has $M \times N$ pixels, the notation for the discretization is then given as:

- The space step: h.
- The time step: Δt.
- The grid points: $(x_i, y_j) = (ih, jh)$, where $1 \leq i \leq M$ and $1 \leq j \leq N$.

The discrete approximation of $\phi(t, x, y)$ is now defined as $\phi^k_{i,j} = \phi(k\Delta t, x_i, y_j)$, where $k \geq 0$, and ϕ^0 is the approximation of the initial curve. The edge detector $g(|\nabla u_0(x, y)|)$ is also discretized approximately by $g_{i,j} = g(|\nabla u_0(x_i, y_j)|)$. Denoting upper indices x and y as the direction of the derivatives, the finite differences are used to approximate the partial derivatives of the gradient:

Backward differences:

$$\Delta^x_- \phi^k_{i,j} = \frac{\phi^k_{i,j} - \phi^k_{i-1,j}}{h}, \qquad \Delta^y_- \phi^k_{i,j} = \frac{\phi^k_{i,j} - \phi^k_{i,j-1}}{h},$$
$$\Delta^x_- g_{i,j} = \frac{g_{i,j} - g_{i-1,j}}{h}, \qquad \Delta^y_- g_{i,j} = \frac{g_{i,j} - g_{i,j-1}}{h}.$$

Forward differences:

$$\Delta^x_+ \phi^k_{i,j} = \frac{\phi^k_{i+1,j} - \phi^k_{i,j}}{h}, \qquad \Delta^y_+ \phi^k_{i,j} = \frac{\phi^k_{i,j+1} - \phi^k_{i,j}}{h}.$$

Central differences:

$$\Delta^x_0 \phi^k_{i,j} = \frac{\phi^k_{i+1,j} - \phi^k_{i-1,j}}{2h}, \qquad \Delta^y_0 \phi^k_{i,j} = \frac{\phi^k_{i,j+1} - \phi^k_{i,j-1}}{2h}.$$

With these differences and using that $\dfrac{\partial \phi}{\partial t} \approx \dfrac{\phi^{k+1}_{i,j} - \phi^k_{i,j}}{\Delta t}$, we can give an

explicit scheme for the discretized Euler-Lagrange equation:

$$\begin{aligned}\frac{\phi_{i,j}^{k+1} - \phi_{i,j}^{k}}{\Delta t} = \delta_\varepsilon\left(\phi_{i,j}^{k}\right)\ \mu\Bigg[&\frac{\Delta_-^x\, g_{i,j}\,\Delta_+^x\,\phi_{i,j}^{k}}{\sqrt{\left(\Delta_+^x\,\phi_{i,j}^{k}\right)^2 + \left(\Delta_0^y\,\phi_{i,j}^{k}\right)^2}} \\ &+ \frac{\Delta_-^y\, g_{i,j}\,\Delta_+^y\,\phi_{i,j}^{k}}{\sqrt{\left(\Delta_0^x\,\phi_{i,j}^{k}\right)^2 + \left(\Delta_+^y\,\phi_{i,j}^{k}\right)^2}} \\ &+ g\,\Delta_-^x\cdot\left(\frac{\Delta_+^x\,\phi_{i,j}^{k}}{\sqrt{\left(\Delta_+^x\,\phi_{i,j}^{k}\right)^2 + \left(\Delta_0^y\,\phi_{i,j}^{k}\right)^2}}\right) \\ &+ g\,\Delta_-^y\cdot\left(\frac{\Delta_+^y\,\phi_{i,j}^{k}}{\sqrt{\left(\Delta_0^x\,\phi_{i,j}^{k}\right)^2 + \left(\Delta_-^y\,\phi_{i,j}^{k}\right)^2}}\right)\Bigg] \\ &- \delta_\varepsilon\left(\phi_{i,j}^{k}\right)\Big[\nu + \lambda_1\left(u_{0,i,j} - c_1(\phi_{i,j}^{k})\right)^2 \\ &\qquad - \lambda_2\left(u_{0,i,j} - c_2(\phi_{i,j}^{k})\right)^2\Big].\end{aligned} \tag{3.8}$$

Algorithm 1 An Active Contour Method

Initialize: $\phi^0 = \phi_0$ and set $k = 0$.

repeat

- Compute $c_1(\phi^k)$ and $c_2(\phi^k)$ by:

$$c_1(\phi^k) \leftarrow \frac{\sum_{i,j} u_{0,i,j}\, H_\varepsilon(\phi_{i,j}^{k})}{\sum_{i,j} H_\varepsilon(\phi_{i,j}^{k})}$$

and

$$c_2(\phi^k) \leftarrow \frac{\sum_{i,j}\left(1 - u_{0,i,j}\, H_\varepsilon(\phi_{i,j}^{k})\right)}{\sum_{i,j}\left(1 - H_\varepsilon(\phi_{i,j}^{k})\right)}.$$

- Obtain ϕ^{k+1} by using the explicit scheme for the Euler-Lagrange equation (3.8).

- Set $k \leftarrow k+1$ if the *until*-condition is not satisfied.

until $\sum_{i,j} |\phi_{i,j}^{k+1} - \phi_{i,j}^{k}| < \Delta t \cdot \epsilon$, where $\epsilon > 0$.

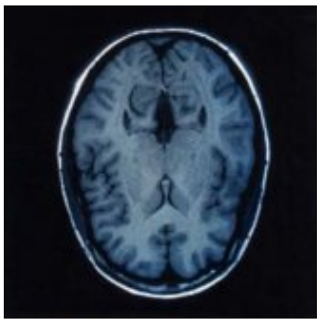
3.4 Experimental Results

The programming tool MATLAB R2012a is used to implement the algorithm above. The edge detector (2.2) is incorporated into the original code written by Y. Wu. All colour images are converted to greyscale images using the matlab function `rgb2gray`. Implementation in MATLAB and `rgb2gray` is also used in the forthcoming chapters.

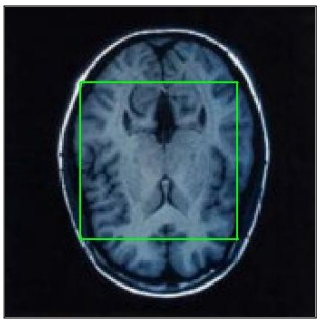
The parameters which are fixed for all experiments, are the time step $\Delta t = 0.5$, $\epsilon = 0.18^2$, the space step $h = 1$, $\lambda_1 = \lambda_2 = 1$, and $a = b = 1$ in the edge detector (2.2).

The initial curve for all images is given as a rectangle placed at the center of the image, where the value inside the curve is 1 and the value outside the curve is 0.

By [2] we know that the active contour method is good, but we want to see how the edge detector influences this method. For a higher value of μ, the result of the segmentation without using edge detector gives either no segmentation or bad segmentation. Thus, using the edge detector, a better segmentation result is produced by choosing a suitable σ.



(a) The original image.

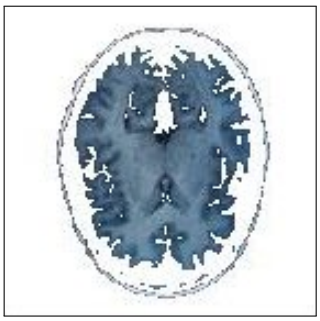


(b) Image with the initial curve.

(c) Segmented result without edge detector.

(d) Segmented result with the original image.

(e) Segmented result with edge detector using $\sigma = 10$.

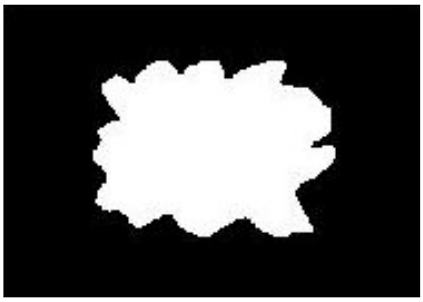


(f) Segmented result with the original image.

Figure 3.3: Segmentation of a brain scan image using $\mu = 1.05 \cdot 255^2$.

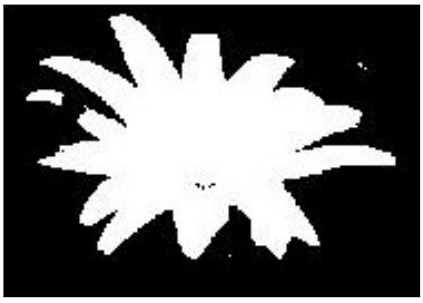
(a) The original image.

(b) Image with the initial curve.



(c) Segmented result without edge detector.

(d) Segmented result with the original image.



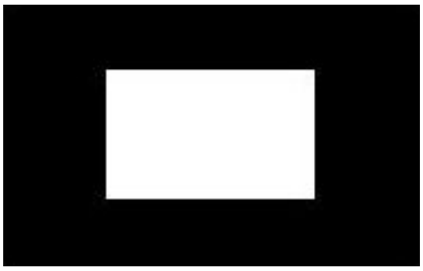
(e) Segmented result with edge detector using $\sigma = 10$.

(f) Segmented result with the original image.

Figure 3.4: Segmentation of a flower image using $\mu = 4 \cdot 255^2$.

(a) The original image.

(b) Image with the initial curve.



(c) Segmented result without edge detector.

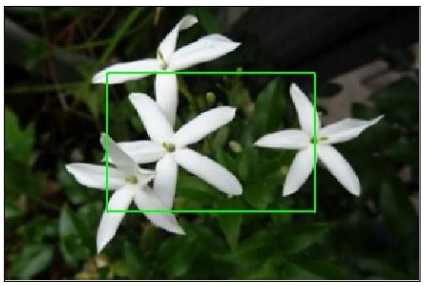
(d) Segmented result with the original image.

(e) Segmented result with edge detector using $\sigma = 15$.

(f) Segmented result with the original image.

Figure 3.5: Segmentation of an image with a set of keys using $\mu = 3.3 \cdot 255^2$.

(a) The original image.

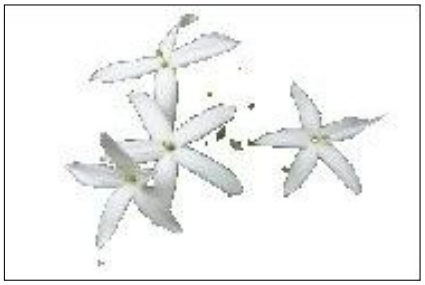


(b) Image with the initial curve.

(c) Segmented result without edge detector.

(d) Segmented result with the original image.

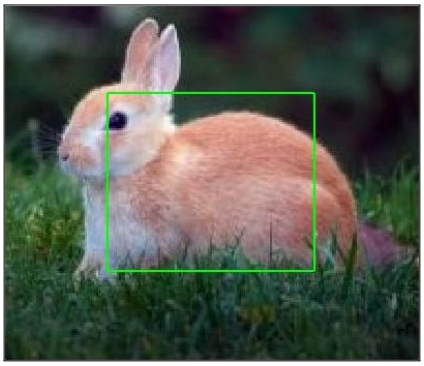
(e) Segmented result with edge detector using $\sigma = 10$.



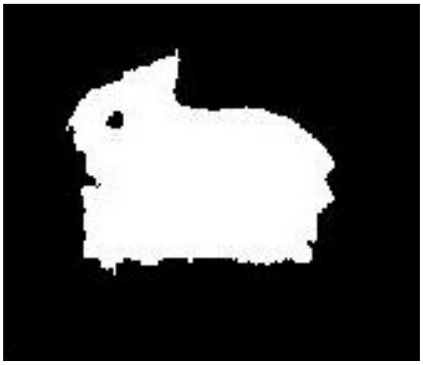
(f) Segmented result with the original image.

Figure 3.6: Segmentation of a Jasmin flower image using $\mu = 4 \cdot 255^2$.

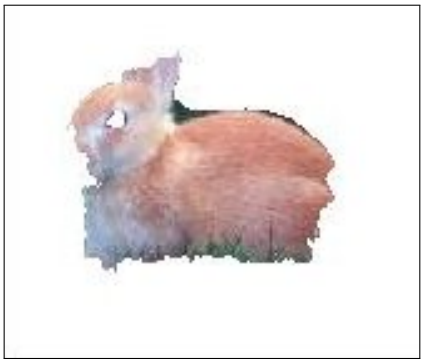
(a) The original image.



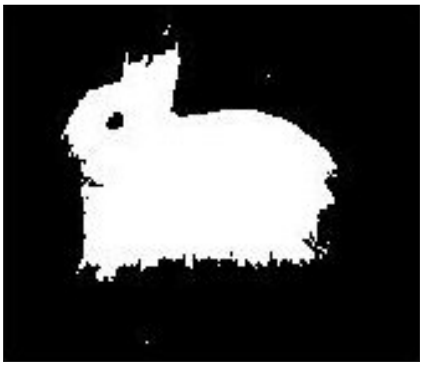
(b) Image with the initial curve.



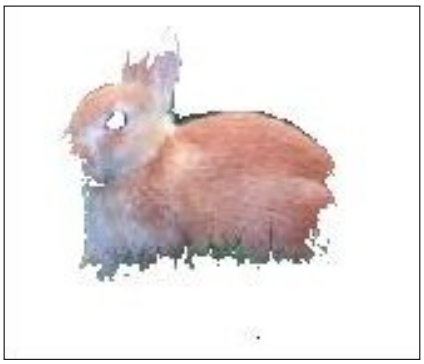
(c) Segmented result without edge detector.



(d) Segmented result with the original image.



(e) Segmented result with edge detector using $\sigma = 10$.



(f) Segmented result with the original image.

Figure 3.7: Segmentation of a rabbit image using $\mu = 3 \cdot 255^2$.

Chapter 4

A Piecewise Constant Level Set Method Applied to Mumford-Shah Image Segmentation

4.1 The Basic Idea of the Model

Assume $u_0 : \Omega \to \mathbb{R}$ is the given input image. The idea of this model is to decompose the domain Ω into a set of disjoint subdomains Ω_i such that $\Omega = \cup_i \Omega_i \cup \Gamma$, where Γ is an interface separating the different subdomains. Using this, we want to find an approximation u of u_0 where u is constant inside Ω_i.

The method in chapter 3, which is based on the Chan-Vese approach, uses level set method and Mumford-Shah segmentation to solve a minimization problem. The interface, which seperates the region of interest and the background, is constructed by level set functions and is expressed implicitly by the zero level set of a Lipschitz continuous function.

Instead of using the zero level set formulation, Lie, Lysaker and Tai [3] propose to represent the interface implicitly by the discontinuities of piecewise constant level set functions. If e.g. an image is divided into two regions, the segmentation is achieved by the level set function ϕ which takes the value 1 in one region and -1 in the other region. Hence, ϕ satisfies the quadratic constraint $\phi^2 = 1$.

This chapter studies the model proposed in [3], which uses a piecewise constant level set model to segment an image by minimizing a constrained problem.

4.2 Level Set Formulation of the Model

Let $u_0 : \Omega \to \mathbb{R}$ be the given image, where Ω is a bounded subset in $\mathbb{R}^2$.

Furthermore we assume that there exists an interface Γ enclosing the subregion $\omega \subset \Omega$. In chapter 3, the standard level set function ϕ was introduced:

$$\begin{cases} \phi(x,y) = 0 & \text{for } (x,y) \in \Gamma, \\ \phi(x,y) > 0 & \text{for } (x,y) \in \text{inside}\,(\Gamma)\,, \\ \phi(x,y) < 0 & \text{for } (x,y) \in \text{outside}\,(\Gamma)\,. \end{cases}$$

In [3], this definition of a level set function ϕ is replaced by

$$\phi(x,y) \;=\; \begin{cases} 1, & \text{if } (x,y) \in \text{int}\,(\omega)\,, \\ -1, & \text{if } (x,y) \in \text{ext}\,(\omega)\,, \end{cases}$$

and the interface Γ is implicitly given by the discontinuity of the function ϕ.

In image segmentation, this idea is used by assuming that the image u_0 consists of two disjoint regions Ω_1 and Ω_2. Now, we construct a piecewise constant approximation u of u_0 by

$$u(x,y) \;=\; \begin{cases} c_1, & \text{if } (x,y) \in \Omega_1, \\ c_2, & \text{if } (x,y) \in \Omega_2, \end{cases} \tag{4.1}$$

where the two constants c_1 and c_2 are the distinct values in Ω_1 and Ω_2, respectively. Letting $\phi = 1$ in Ω_1 and $\phi = -1$ in Ω_2, u can be written as:

$$u \;=\; \frac{1}{2}\Big[\, c_1\,(\phi + 1) \;-\; c_2\,(\phi - 1)\,\Big].$$

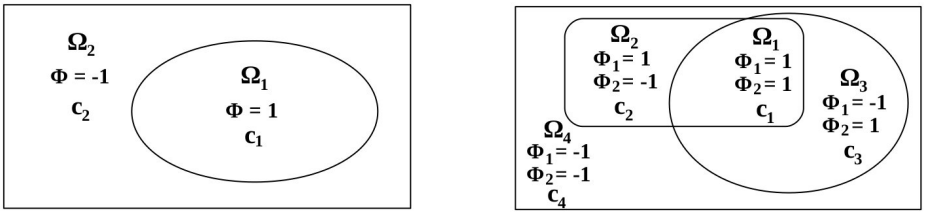
Using the two level set function ϕ_1 and ϕ_2, the piecewise constant function u can be constructed by four constants c_1, c_2, c_3 and c_4, representing the distinct values in Ω_1, Ω_2, Ω_3, and Ω_4, respectively:

$$u(x,y) \;=\; \begin{cases} c_1, & \text{if } \phi_1(x,y) = \;\;1, \;\; \phi_2(x,y) = \;\;1, \\ c_2, & \text{if } \phi_1(x,y) = \;\;1, \;\; \phi_2(x,y) = -1, \\ c_3, & \text{if } \phi_1(x,y) = -1, \;\; \phi_2(x,y) = \;\;1, \\ c_4, & \text{if } \phi_1(x,y) = -1, \;\; \phi_2(x,y) = -1, \end{cases} \tag{4.2}$$

and can also be written as:

$$\begin{aligned} u \;=\; \frac{1}{4}\Big[& c_1\,(\phi_1 + 1)(\phi_2 + 1) \;-\; c_2\,(\phi_1 + 1)(\phi_2 - 1) \\ & -\; c_3\,(\phi_1 - 1)(\phi_2 + 1) \;+\; c_4\,(\phi_1 - 1)(\phi_2 - 1)\Big]. \end{aligned} \tag{4.3}$$

We see that the two level set functions $\{\phi_j\}_{j=1}^{2}$ are satisfying the constraints $\phi_j^2 = 1$, for $j = 1, 2$. The idea of (4.1) and (4.2) is shown in fig. 4.1.



(a) Illustrating (4.1) where $u(x, y)$ is constructed by one level set function.

(b) Illustrating (4.2) where $u(x, y)$ is constructed by two level set functions.

Figure 4.1

We simplify equation (4.3) by introducing the basis functions ψ_i :

$$\begin{aligned} u &= \frac{1}{4}\Big[c_1 \underbrace{(\phi_1+1)(\phi_2+1)}_{=:\,4\cdot\psi_1} - c_2 \underbrace{(\phi_1+1)(\phi_2-1)}_{=:\,-4\cdot\psi_2} \\ &\qquad - c_3 \underbrace{(\phi_1-1)(\phi_2+1)}_{=:\,-4\cdot\psi_3} + c_4 \underbrace{(\phi_1-1)(\phi_2-1)}_{=:\,4\cdot\psi_4} \Big]. \\ &= \sum_{i=1}^{4} c_i\, \psi_i(\boldsymbol{\phi}), \end{aligned}$$

where $\psi_i(\boldsymbol{\phi}) = \psi_i(\phi_1, \phi_2)$.

We generalize this idea for a general piecewise constant function u with n level set functions. For the generalization, we need to seperate the image into 2^n regions. Each level set function $\{\phi_j\}_{j=1}^{n}$ must satisfy the constraint $\phi_j^2 = 1$. If this is the case, the 2^n constants, $\{c_i\}_{i=1}^{2^n}$, need to be chosen.

The function u can now be written as the weighted sum:

$$u = \sum_{i=1}^{2^n} c_i\, \psi_i, \tag{4.4}$$

where

$$\begin{aligned} \psi_i := \ \psi_i(\boldsymbol{\phi}) \ &= \ \psi_i(\phi_1, \ldots, \phi_n) \\ &= \ \frac{(-1)^{s(i)}}{2^n} \prod_{j=1}^{n} (\phi_j + 1 - 2\, b_j^{i-1}), \\ s(i) \ &= \ \sum_{j=1}^{n} b_j^{i-1}, \end{aligned}$$

and $(b_1^{i-1}, b_2^{i-1}, \ldots, b_n^{i-1})$ is the binary representation of $i-1$.

If u can be written as above, we obtain

$$\begin{aligned} \operatorname{supp}(\psi_i) &= \Omega_i, \\ \cup_i \operatorname{supp}(\psi_i) &= \Omega, \\ \operatorname{supp}(\psi_i) \cap \operatorname{supp}(\psi_l) &= \emptyset, \quad \text{when } i \neq l, \end{aligned}$$

for $i = 1, \ldots, 2^n$, where

$$\psi_i(\boldsymbol{\phi}(x,y)) = \begin{cases} 1, & \text{if } (x,y) \in \operatorname{int}(\Omega_i) \\ 0, & \text{if } (x,y) \in \operatorname{ext}(\Omega_i) \end{cases}$$

Since the level set functions $\boldsymbol{\phi}$ satisfy $\phi_j^2 = 1$ for $j = 1, \ldots, n$, we can use the basis functions ψ_i to find the length of the boundary of Ω_i and the area inside Ω_i :

$$\begin{aligned} \operatorname{Length}(\partial\Omega_i) &= \int_\Omega |\nabla\psi_i(\boldsymbol{\phi}(x,y))| \, dxdy, \\ \operatorname{Area}(\Omega_i) &= \int_\Omega \psi_i(\boldsymbol{\phi}(x,y)) \, dxdy. \end{aligned}$$

In this chapter, as in the last chapter, we use instead a weighted length term:

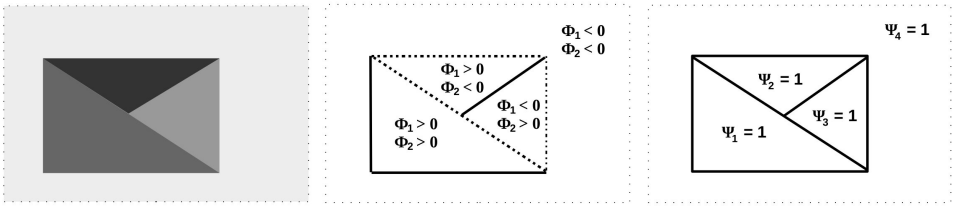
$$\operatorname{Length}_g(\partial\Omega_i) = \int_\Omega g(|\nabla u_0(x,y)|) \, |\nabla\psi_i(\boldsymbol{\phi}(x,y))| \, dxdy,$$

where $g(|\nabla u_0|)$ is the edge detector (2.2). To use the length term in numerical computations, the gradient is approximated by the finite differences $(\psi_i)_x$ and $(\psi_i)_y$:

$$\operatorname{Length}_g(\partial\Omega_i) = \int_\Omega g(|\nabla u_0|) \sqrt{(\psi_i)_x^2 + (\psi_i)_y^2 + \epsilon} \, dxdy, \tag{4.5}$$

for a small $\epsilon > 0$.

The length term used here is more correct than using the regularizer term in chapter 3, which is given as $\int_\Omega g(|\nabla u_0|) \, \delta_\varepsilon(\phi_j) \, |\nabla\phi_j| \, dxdy$. The advantage of (4.5) is that it counts every edge twice and therefore treats all edges equally, while the length term of chapter 3 counts some egdes once and some edges twice. The result of this is that some edges are relatively weighted more important than others. Fig. 4.2 shows an illustration of the different terms.



(a) (b) (c)

Figure 4.2: **(a)** A simple image example with four phases. **(b)** Measuring the length of the boundary with the regularizer used in chapter 3, which counts the thick dashed edges once and the thick continuous edges twice. **(c)** The length term of this chapter measures all the edges, which is shown by the thick lines, twice.

Relating the Model with the Mumford Shah Functional

Let u be a piecewise constant approximation of the image $u_0 : \Omega \to \mathbb{R}$ and let Γ be given as the interface between the regions Ω_i, for $i = 1, \ldots, 2^n$. Then the Mumford-Shah functional [5] can be written as:

$$E^{MS}(u, \Gamma) = \beta \cdot \text{Length}(\Gamma) + \int_{\Omega \setminus \Gamma} |\nabla u(x, y)|^2 \, dxdy + \zeta \cdot \int_{\Omega} |u(x, y) - u_0(x, y)|^2 \, dxdy,$$

where the first term is the regularization term measuring the total length of the interface, the second term controls the smoothness of u inside $\Omega \setminus \Gamma$, and the third term measures how well u approximates u_0. The fixed positive parameters β and ζ control the influence of the first and the third term, respectively.

In our case the second term disappears because the piecewise constant image u is constant inside each Ω_i.

The total length of the interface between the regions can now be obtained by summing all parts of the interface:

$$\text{Length}_g(\Gamma) = \sum_{i=1}^{2^n} \int_{\Omega} g(|\nabla u_0(x, y)|) \, |\nabla \psi_i(\boldsymbol{\phi}(x, y))| \, dxdy,$$

where Γ is the curve separating the different regions Ω_i.

The reduced form of the Mumford-Shah functional is now given by:

$$E(\boldsymbol{\phi}, \boldsymbol{c}) = \frac{1}{2} \int_{\Omega} |u - u_0|^2 \, dxdy + \beta \sum_{i=1}^{2^n} \int_{\Omega} g \, |\nabla \psi_i| \, dxdy,$$

where u is given as in (4.4) and $\beta \geq 0$.

As $\phi_j^2 = 1$, we obtain the following optimization problem, where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$ and $\boldsymbol{c} = (c_1, \ldots, c_{2^n})$:

$$\begin{aligned} \min_{\boldsymbol{\phi},\, \boldsymbol{c}} \quad & E(\boldsymbol{\phi}, \boldsymbol{c}) \\ \text{subject to} \quad & \phi_j^2 - 1 = 0, \quad \text{for all } j = 1, \ldots, n. \end{aligned} \tag{4.6}$$

Now, the problem is to find the local minimizers $\boldsymbol{\phi}$ and $\boldsymbol{c}$ of $E(\boldsymbol{\phi}, \boldsymbol{c})$ while satifying the constraint in (4.6). This problem can be solved by a Projection Lagrangian approach [11, 17]. Therefore we introduce the Lagrangian functional:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi},\, \boldsymbol{c},\, \boldsymbol{\lambda}) &= E(\boldsymbol{\phi}, \boldsymbol{c}) \; + \sum_{j=1}^{n} \int_\Omega \lambda_j (\phi_j^2 - 1)\, dxdy \\ &= \frac{1}{2} \int_\Omega |u - u_0|^2 \, dxdy \; + \; \beta \sum_{i=1}^{2^n} \int_\Omega g\, |\nabla \psi_i| \, dxdy \\ &\quad + \sum_{j=1}^{n} \int_\Omega \lambda_j (\phi_j^2 - 1)\, dxdy \\ &= \frac{1}{2} \int_\Omega \Big| \sum_{i=1}^{2^n} c_i\, \psi_i - u_0 \Big|^2 \, dxdy \; + \; \beta \sum_{i=1}^{2^n} \int_\Omega g\, |\nabla \psi_i| \, dxdy \\ &\quad + \sum_{j=1}^{n} \int_\Omega \lambda_j (\phi_j^2 - 1)\, dxdy \end{aligned}$$

where $\lambda_1, \ldots, \lambda_n$ are the Lagrangian multipliers denoted by the set $\boldsymbol{\lambda} = \{\lambda_j\}_{j=1}^n$. The minimizer of the optimization problem (4.6) can be found by searching for the saddle point of $\mathcal{L}(\boldsymbol{\phi},\, \boldsymbol{c},\, \boldsymbol{\lambda})$. At this point we need to have:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \phi_j} = 0, & j = 1, \ldots, n\,, \\ \dfrac{\partial \mathcal{L}}{\partial c_i} = 0, & i = 1, \ldots, 2^n\,, \\ \dfrac{\partial \mathcal{L}}{\partial \lambda_j} = 0, & j = 1, \ldots, n\,. \end{cases}$$

The saddle point is obtained by minimizing $\mathcal{L}$ w.r.t. $\boldsymbol{\phi}$ and $\boldsymbol{c}$ and maximizing $\mathcal{L}$ w.r.t. $\boldsymbol{\lambda}$. We see later that maximizing $\mathcal{L}$ w.r.t $\boldsymbol{\lambda}$ implies that the constraints are satisfied, and then the last term of $\mathcal{L}(\boldsymbol{\phi},\, \boldsymbol{c},\, \boldsymbol{\lambda})$ vanishes such that $\mathcal{L} = E$.

Minimizing $\mathcal{L}$ w.r.t. $\boldsymbol{\phi}$:

Consider the minimization of $\mathcal{L}$ with respect to ϕ_j for $j = 1, \ldots, n$. We denote the Lagrangian functional $\mathcal{L}(\boldsymbol{\phi},\ \boldsymbol{c},\ \boldsymbol{\lambda})$ as $\mathcal{L}(\phi_j)$ and the piecewise constant level set functions $\psi_i(\boldsymbol{\phi})$ as $\psi_i(\phi_j)$, for $i = 1, \ldots, 2^n$.

Let us evaluate the Gâteaux differential [26, p. 23] of the functional $\mathcal{L}$ with respect to ϕ_j in the direction η_j :

$$
\begin{aligned}
D_{\eta_j}\mathcal{L}(\phi_j) &= \lim_{\tau\to 0}\frac{1}{\tau}\big[\mathcal{L}(\phi_j+\tau\,\eta_j)\ -\ \mathcal{L}(\phi_j)\big] \\
&= \frac{1}{2}\ \lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega\left[\Big|\sum_{i=1}^{2^n} c_i\,\psi_i(\phi_j+\tau\,\eta_j)-u_0\Big|^2-\Big|\sum_{i=1}^{2^n} c_i\,\psi_i(\phi_j)-u_0\Big|^2\right]dxdy \\
&\quad+\beta\ \lim_{\tau\to 0}\frac{1}{\tau}\sum_{i=1}^{2^n}\int_\Omega g\,\Big[|\nabla\psi_i(\phi_j+\tau\,\eta_j)|-|\nabla\psi_i(\phi_j)|\,\Big]dxdy \\
&\quad+\ \lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega\Big[\lambda_j\big((\phi_j+\tau\,\eta_j)^2-\phi_j^2\big)\Big]\,dxdy.
\end{aligned}
$$

We calculate the limit of the first part of $D_{\eta_j}\mathcal{L}(\phi_j)$, by using that $|\vec{a}|^2-|\vec{b}|^2 = (\vec{a}-\vec{b})\cdot(\vec{a}+\vec{b})$ and L'Hopital's rule, such that:

$$
\begin{aligned}
&\frac{1}{2}\ \lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega\left[\Big|\sum_{i=1}^{2^n} c_i\,\psi_i(\phi_j+\tau\,\eta_j)-u_0\Big|^2-\Big|\sum_{i=1}^{2^n} c_i\,\psi_i(\phi_j)-u_0\Big|^2\right]dxdy \\
&\quad=\frac{1}{2}\ \lim_{\tau\to 0}\int_\Omega\frac{1}{\tau}\Big[\sum_{i=1}^{2^n}\big(c_i\,\psi_i(\phi_j+\tau\,\eta_j)\ -\ c_i\,\psi_i(\phi_j)\big)\Big] \\
&\qquad\qquad\cdot\Big[\sum_{i=1}^{2^n}\big(c_i\,\psi_i(\phi_j+\tau\,\eta_j)\ +\ c_i\,\psi_i(\phi_j)\big)\ -\ 2\,u_0\Big]\,dxdy \\
&\quad=\frac{1}{2}\ \lim_{\tau\to 0}\int_\Omega 2\,\Big(\sum_{i=1}^{2^n} c_i\,\frac{\partial\psi_i(\phi_j+\tau\,\eta_j)}{\partial\phi_j}\eta_j\Big)\Big[\sum_{i=1}^{2^n} c_i\,\psi_i(\phi_j+\tau\,\eta_j)\ -\ u_0\Big]\,dxdy \\
&\quad=\int_\Omega\Big(\sum_{i=1}^{2^n} c_i\,\frac{\partial\psi_i}{\partial\phi_j}\eta_j\Big)\Big[\sum_{i=1}^{2^n} c_i\,\psi_i\ -\ u_0\Big]\,dxdy \\
&\quad=\int_\Omega(u\ -\ u_0)\sum_{i=1}^{2^n} c_i\,\frac{\partial\psi_i}{\partial\phi_j}\eta_j\,dxdy.
\end{aligned}
$$

The limit of the second part of $D_{\eta_j}\mathcal{L}(\phi_j)$ is calculated by using analogously

$(\vec{a}-\vec{b})\cdot(\vec{a}+\vec{b}) = |\vec{a}|^2 - |\vec{b}|^2$, the Taylor expansion $\lim\limits_{h\to 0}\frac{1}{h}\Big[f(\vec{a}+h\cdot\vec{b}) - f(\vec{a})\Big]$ $= f'(\vec{a})\cdot\vec{b}$, and the Divergence theorem:

$$
\begin{aligned}
&\lim_{\tau\to 0}\frac{1}{\tau}\sum_{i=1}^{2^n}\int_\Omega g\Big[|\nabla\psi_i(\phi_j+\tau\,\eta_j)| - |\nabla\psi_i(\phi_j)|\Big]dxdy\\
&= \sum_{i=1}^{2^n}\lim_{\tau\to 0}\int_\Omega g\,\frac{\big(\nabla\psi_i(\phi_j+\tau\,\eta_j)+\nabla\psi_i(\phi_j)\big)}{|\nabla\psi_i(\phi_j+\tau\,\eta_j)|+|\nabla\psi_i(\phi_j)|}\cdot\frac{\big(\nabla\psi_i(\phi_j+\tau\,\eta_j)-\nabla\psi_i(\phi_j)\big)}{\tau}\,dxdy\\
&= \sum_{i=1}^{2^n}\lim_{\tau\to 0}\int_\Omega g\,\frac{\big(\nabla\psi_i(\phi_j+\tau\,\eta_j)+\nabla\psi_i(\phi_j)\big)}{|\nabla\psi_i(\phi_j+\tau\,\eta_j)|+|\nabla\psi_i(\phi_j)|}\cdot\nabla\Big(\frac{\psi_i(\phi_j+\tau\,\eta_j)-\psi_i(\phi_j)}{\tau}\Big)\,dxdy\\
&= \sum_{i=1}^{2^n}\int_\Omega g\,\frac{\nabla\psi_i(\phi_j)}{|\nabla\psi_i(\phi_j)|}\cdot\nabla\Big(\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\Big)\,dxdy\\
&= -\sum_{i=1}^{2^n}\int_\Omega\nabla\cdot\Big(g\,\frac{\nabla\psi_i}{|\nabla\psi_i|}\Big)\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\,dxdy + \sum_{i=1}^{2^n}\int_{\partial\Omega}\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\Big(g\,\frac{\nabla\psi_i}{|\nabla\psi_i|}\cdot\vec{n}\Big)\,ds\\
&= -\sum_{i=1}^{2^n}\int_\Omega\frac{\nabla g\cdot\nabla\psi_i}{|\nabla\psi_i|}\,\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\,dxdy - \sum_{i=1}^{2^n}\int_\Omega g\,\nabla\cdot\Big(\frac{\nabla\psi_i}{|\nabla\psi_i|}\Big)\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\,dxdy\\
&\quad + \sum_{i=1}^{2^n}\int_{\partial\Omega}\frac{\partial\psi_i}{\partial\phi_j}\,\eta_j\Big(g\,\frac{\nabla\psi_i}{|\nabla\psi_i|}\cdot\vec{n}\Big)\,ds,
\end{aligned}
$$

where $\vec{n}$ is the outward normal vector to the boundary $\partial\Omega$.

Now we consider the limit of the last part of $D_{\eta_j}\mathcal{L}(\phi_j)$:

$$
\begin{aligned}
&\lim_{\tau\to 0}\frac{1}{\tau}\int_\Omega\Big[\lambda_j\big((\phi_j+\tau\,\eta_j)^2-\phi_j^2\big)\Big]\,dxdy\\
&\qquad = \lim_{\tau\to 0}\int_\Omega\lambda_j\,\frac{(\phi_j+\tau\,\eta_j\,-\,\phi_j)(\phi_j+\tau\,\eta_j\,+\,\phi_j)}{\tau}\,dxdy\\
&\qquad = \lim_{\tau\to 0}\int_\Omega\lambda_j\,\frac{\tau\,\eta_j\,(2\phi_j+\tau\,\eta_j)}{\tau}\,dxdy\\
&\qquad = \lim_{\tau\to 0}\int_\Omega\lambda_j\,\eta_j\,(2\,\phi_j+\tau\,\eta_j)\,dxdy\\
&\qquad = \int_\Omega 2\,\lambda_j\,\phi_j\,\eta_j\,dxdy.
\end{aligned}
$$

Summarizing all three limits, we obtain:

$$
\begin{aligned}
D_{\eta_j}\mathcal{L}(\phi_j) = &\int_\Omega (u - u_0) \sum_{i=1}^{2^n} c_i \frac{\partial \psi_i}{\partial \phi_j} \eta_j \, dxdy \\
&- \beta \sum_{i=1}^{2^n} \int_\Omega \frac{\nabla g \cdot \nabla \psi_i(\phi_j)}{|\nabla \psi_i(\phi_j)|} \frac{\partial \psi_i}{\partial \phi_j} \eta_j \, dxdy \\
&- \beta \sum_{i=1}^{2^n} \int_\Omega g \, \nabla \cdot \Big(\frac{\nabla \psi_i(\phi_j)}{|\nabla \psi_i(\phi_j)|} \Big) \frac{\partial \psi_i}{\partial \phi_j} \eta_j \, dxdy \\
&+ \beta \sum_{i=1}^{2^n} \int_{\partial\Omega} \frac{\partial \psi_i}{\partial \phi_j} \eta_j \Big(g \frac{\nabla \psi_i(\phi_j)}{|\nabla \psi_i(\phi_j)|} \cdot \vec{n} \Big) \, ds \\
&+ \int_\Omega 2 \lambda_j \phi_j \eta_j \, dxdy.
\end{aligned}
$$

By the definition of the Gâteaux differential, it follows that

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \phi_j} &= \frac{\partial E}{\partial \phi_j} + 2 \lambda_j \phi_j \\
&= (u - u_0) \sum_{i=1}^{2^n} c_i \frac{\partial \psi_i}{\partial \phi_j} \\
&\quad - \beta \sum_{i=1}^{2^n} \left(\frac{\nabla g \cdot \nabla \psi_i(\phi_j)}{|\nabla \psi_i(\phi_j)|} + g \, \nabla \cdot \Big(\frac{\nabla \psi_i(\phi_j)}{|\nabla \psi_i(\phi_j)|} \Big) \right) \frac{\partial \psi_i}{\partial \phi_j} \\
&\quad + 2 \lambda_j \phi_j.
\end{aligned}
$$

Minimizing $\mathcal{L}$ w.r.t. c :

We know that only the first term of $\mathcal{L}$ depends on u and since $u = \sum_{i=1}^{2^n} c_i \psi_i$, we obtain, by applying the chain rule, as done in [10], that:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_i} &= \frac{\partial}{\partial c_i} \Bigg[\frac{1}{2} \int_\Omega \Big| \sum_{l=1}^{2^n} c_l \psi_l - u_0 \Big|^2 \, dxdy + \beta \sum_{l=1}^{2^n} \int_\Omega g \, |\nabla \psi_l| \, dxdy \\
&\qquad + \sum_{j=1}^{n} \int_\Omega \lambda_j (\phi_j^2 - 1) \, dxdy \Bigg] \\
&= \int_\Omega \Big(\sum_{l=1}^{2^n} c_l \psi_l - u_0 \Big) \psi_i \, dxdy \\
&= \int_\Omega (u - u_0) \, \psi_i \, dxdy.
\end{aligned}
$$

Maximizing $\mathcal{L}$ w.r.t. $\boldsymbol{\lambda}$:

We regain the constraint when differentiating $\mathcal{L}$ w.r.t. λ_j, for $j = 1, \ldots, n$:

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = \phi_j^2 - 1.$$

A Numerical Method of the Model

The Projection Lagrangian approach has the condition that the derivatives $\frac{\partial \mathcal{L}}{\partial \phi_j}$, $\frac{\partial \mathcal{L}}{\partial c_i}$ and $\frac{\partial \mathcal{L}}{\partial \lambda_j}$, where $i = 1, \ldots, n$ and $j = 1, \ldots, 2^n$, must be equal to zero at a saddle point of $\mathcal{L}$. This can be approximated by an iterative algorithm. After choosing the initial guesses $\boldsymbol{\phi}^0$, $\boldsymbol{c}^0$ and $\boldsymbol{\lambda}^0$, the overall idea is to repeat a procedure such that better estimates of $\boldsymbol{\phi}^k$, $\boldsymbol{c}^k$ and $\boldsymbol{\lambda}^k$ are achieved. We use $\boldsymbol{\phi}^k$, $\boldsymbol{c}^k$ and $\boldsymbol{\lambda}^k$ to find the next iteration $\boldsymbol{\phi}^{k+1}$, $\boldsymbol{c}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$. After a finite number of steps, the estimates are not varying anymore, which implies that we are at a saddle point. Finding numerically when the derivatives are equal to zero, is done by using three different schemes.

The scheme for $\boldsymbol{\phi}^k$:

We introduce an artificial time variable and search for a steady-state solution of the PDE:

$$\boldsymbol{\phi}_t = -\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}}.$$

It follows that if we minimize $\mathcal{L}$ w.r.t. $\boldsymbol{\phi}$, then the steady-state $\boldsymbol{\phi}_t = 0$ is obtained. Now, we can discretize $\boldsymbol{\phi}_t$ by using the forward Euler method,

$$\boldsymbol{\phi}_t = \frac{\boldsymbol{\phi}^{\text{next}} - \boldsymbol{\phi}^{\text{curr}}}{\Delta t},$$

where $\Delta t > 0$ is a small time step. If we combine both the equations above, we obtain a gradient descent method:

$$\phi_j^{\text{next}} = \phi_j^{\text{curr}} - \Delta t \frac{\partial \mathcal{L}}{\partial \phi_j} \left(\boldsymbol{\phi}^{\text{curr}}, \boldsymbol{c}^k, \boldsymbol{\lambda}^k \right).$$

If we iterate this method infinitely many times, ϕ_j^{next} converges to the exact minimizer of $\mathcal{L}$ w.r.t. $\boldsymbol{\phi}$. Since we need an approximation of this minimizer, we only let the method iterate a fixed number of times. In the numerical implementation the fixed number is 10. After this procedure, we set $\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^{\text{next}}$.

The scheme for $\boldsymbol{c}^k$:

We minimize $\mathcal{L}$ w.r.t. $\boldsymbol{c}$ under the assumption that $u = \sum_{i=1}^{2^n} c_i\,\psi_i$ and

$$\frac{\partial \mathcal{L}}{\partial c_i} \;=\; \int_\Omega (u - u_0)\,\psi_i \; dxdy \;=\; \langle\, u \;-\; u_0, \psi_i \,\rangle_{L^2(\Omega)}\,,$$

where $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ is the L^2 inner product on the region Ω. As a consequence, we can use that

$$\frac{\partial \mathcal{L}}{\partial c_i} \;=\; 0, \tag{4.7}$$

for $i \;=\; 1, \ldots, 2^n$. Using u together with (4.7) and knowing that $\mathcal{L}$ is a quadratic functional in $\boldsymbol{c}$, we obtain a $2^n \times 2^n$ linear system:

$$\boldsymbol{A}\,\boldsymbol{c} \;=\; \boldsymbol{b},$$

where

$$\boldsymbol{A}_{i,l} = \langle\, \psi_i, \psi_l \,\rangle_{L^2(\Omega)} \qquad \text{and} \qquad \boldsymbol{b}_l \;=\; \langle\, u_0, \psi_l \,\rangle_{L^2(\Omega)}.$$

The first step in the discretization of this process is to construct u such that

$$u\left(\boldsymbol{c}^k, \boldsymbol{\phi}^{k+1}\right) \;=\; \sum_{i=1}^{2^n} c_i^k\,\psi_i^{k+1}.$$

After this step, we solve the linear system

$$\sum_{i=1}^{2^n} \langle\, \psi_i^{k+1}, \psi_l^{k+1} \,\rangle_{L^2(\Omega)}\; c_i^{k+1} \;=\; \langle\, u_0, \psi_l^{k+1} \,\rangle_{L^2(\Omega)}\,,$$

for $l = 1, \ldots, 2^n$. A matrix inversion of the linear system should not be done before $|\phi_j^{k+1}| \approx 1\,\forall j$, otherwise the inversion becomes ill-conditioned.

The scheme for $\boldsymbol{\lambda}^k$:

To make a scheme for $\boldsymbol{\lambda}$ we combine $\dfrac{\partial \mathcal{L}}{\partial \lambda_j} \;=\; 0$ and $\dfrac{\partial \mathcal{L}}{\partial \lambda_j} \;=\; \phi_j^2 - 1$ and get $\phi_j^2 \;=\; 1$. Furthermore, by combining $\dfrac{\partial \mathcal{L}}{\partial \phi_j} \;=\; 0$ and $\dfrac{\partial \mathcal{L}}{\partial \phi_j} \;=\; \dfrac{\partial E}{\partial \phi_j} + 2\,\lambda_j\,\phi_j$ we obtain $\lambda_j\,\phi_j = -\,\dfrac{1}{2}\,\dfrac{\partial E}{\partial \phi_j}$. Now, by using both results, we get

$$\lambda_j = -\,\frac{1}{2}\,\phi_j\,\frac{\partial E}{\partial \phi_j}\,.$$

From this, it is possible to propose an updating scheme for $\boldsymbol{\lambda}$:

$$\lambda_j^{k+1} = -\frac{1}{2}\,\phi_j^{k+1}\,\frac{\partial E}{\partial \phi_j}\left(\boldsymbol{\phi}^{k+1},\, \boldsymbol{c}^{k+1},\, \boldsymbol{\lambda}^{k}\right) .$$

Algorithm 2 A Projection Lagrangian Method

Initialize: $\boldsymbol{\phi}^0$, $\boldsymbol{c}^0$, $\boldsymbol{\lambda}^0$ and set $k = 0$.

repeat

- Use gradient descent method to approximate $\boldsymbol{\phi}^{\text{next}}$. Then, set

$$\boldsymbol{\phi}^{k+1} \leftarrow \boldsymbol{\phi}^{\text{next}}.$$

- Construct u by $\boldsymbol{c}^k$ and $\boldsymbol{\phi}^{k+1}$:

$$u \leftarrow \sum_{i=1}^{2^n} c_i^k\, \psi_i^{k+1}.$$

- **if** $|\phi_j^{k+1}| \approx 1$, $\forall j$

  then

  Solve the linear system

$$\sum_{i=1}^{2^n} \left\langle \psi_i^{k+1}, \psi_l^{k+1} \right\rangle_{L^2(\Omega)} c_i^{k+1} \;=\; \left\langle u_0, \psi_l^{k+1} \right\rangle_{L^2(\Omega)} ,$$

  for $l = 1, \ldots, 2^n$.

  else

  $\boldsymbol{c}^{k+1} \leftarrow \boldsymbol{c}^k$.

- Update the Lagrangian multipliers:

$$\lambda_j^{k+1} \leftarrow -\frac{1}{2}\,\phi_j^{k+1}\,\frac{\partial E}{\partial \phi_j}\left(\boldsymbol{\phi}^{k+1},\, \boldsymbol{c}^{k+1},\, \boldsymbol{\lambda}^{k}\right) ,$$

  where $j = 1, \ldots, n$.

- Set $k \leftarrow k + 1$ if the *until*-condition is not satisfied.

until $\| \boldsymbol{\phi}^{k+1} - \boldsymbol{\phi}^k \| < error$.

4.3 Experimental Results

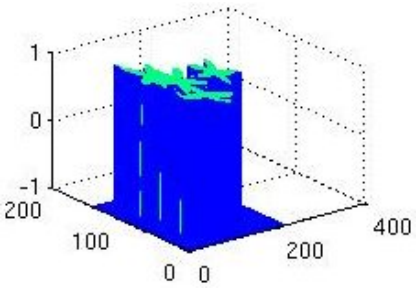
The implementation code for this method is written by J.Lie, but the edge detector (2.2) is incorporated into this code in MATLAB.

By applying the piecewise constant level set method we use the function ϕ to obtain the segmented image. We start with $\phi^0 \equiv 0$ and the initial vector of $\boldsymbol{c} = [0.1, 1]$. In the implementation, the term $\dfrac{\nabla\psi_i}{|\nabla\psi_i|}$ is replaced by $\dfrac{\nabla\psi_i}{\sqrt{|\nabla\psi_i|^2+\epsilon}}$, where we choose $\epsilon \approx 10^{-3}$.

The numerical scheme solves the associated PDE explicitly. Hence, the CFL stability condition might be violated unless the value of β is small. A larger value of β requires a smaller time step value and a small value of β require a larger time step value. Using a small time step increases the computational time significantly. Therefore, we use $\beta = 10^{-6}$ such that the time step is given by $\Delta t = 0.01$.

All the experiments presented below uses the edge detector (2.2) with $\sigma = 1$, $a = 1$, $b = 50$, and the error in the *until*-condition is given by $error = 0.01$.

(a) The original image.

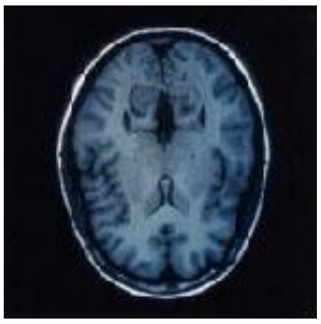


(b) The associated ϕ function.

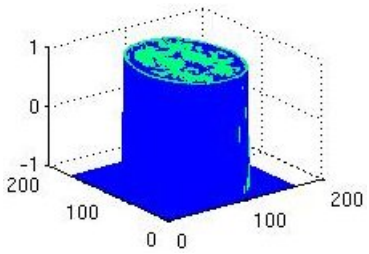
(c) Segmented result.

(d) Segmented result with the original image.

Figure 4.3: Segmentation of an image with Jasmin flowers.



(a) The original image.



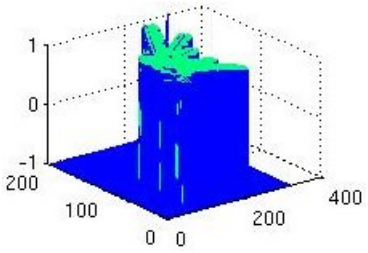
(b) The associated ϕ function.

(c) Segmented result.

(d) Segmented result with the original image.

Figure 4.4: Segmentation of a brain scan image.

(a) The original image.



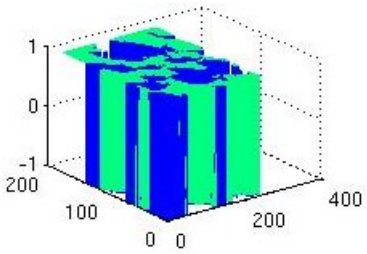
(b) The associated ϕ function.

(c) Segmented result.

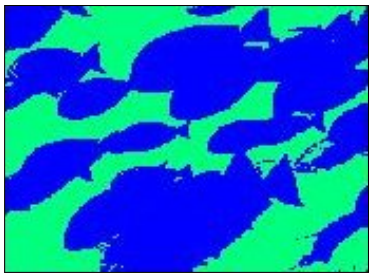
(d) Segmented result with the original image.

Figure 4.5: Segmentation of an image of a set of keys.

(a) The original image.



(b) The associated ϕ function.

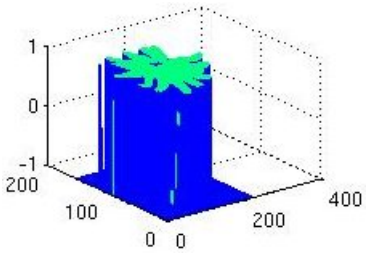


(c) Segmented result.

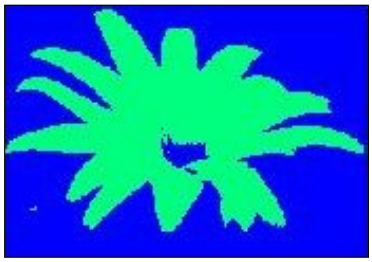
(d) Segmented result with the original image.

Figure 4.6: Segmentation of a fish shoal image.

(a) The original image.



(b) The associated ϕ function.

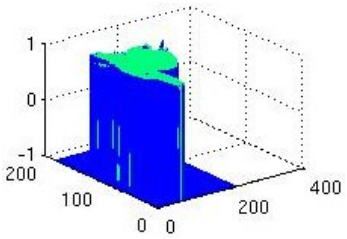


(c) Segmented result.

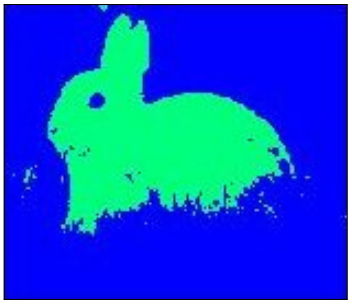
(d) Segmented result with the original image.

Figure 4.7: Segmentation of a flower image.

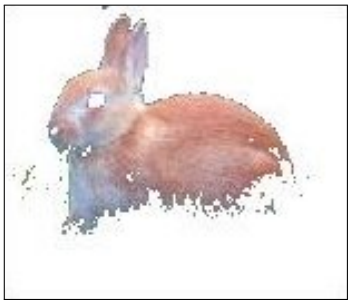
(a) The original image.



(b) The associated ϕ function.



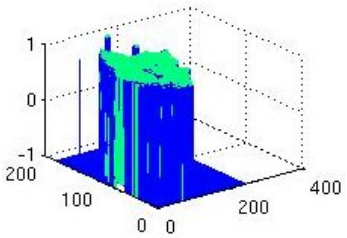
(c) Segmented result.



(d) Segmented result with the original image.

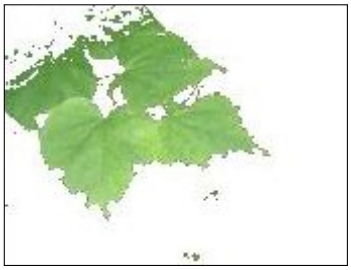
Figure 4.8: Segmentation of a rabbit image.

(a) The original image.



(b) The associated ϕ function.

(c) Segmented result.



(d) Segmented result with the original image.

Figure 4.9: Segmentation of an image of leaves.

Chapter 5

A Spatially Continuous Max-Flow and Min-Cut Framework for Image Segmentation

5.1 The Basic Idea of the Model

We partition an image $u_0 : \Omega \to \mathbb{R}$, with $\Omega \subset \mathbb{R}^2$, into two regions; a foreground region ω and a background region $\Omega \setminus \omega$. The basic idea of the model is to do this by using a variational approach by minimizing the problem

$$\min_{\omega} \int_{\Omega \setminus \omega} C_s(x,y)\,dxdy + \int_{\omega} C_t(x,y)\,dxdy + g\,|\partial\omega|. \tag{5.1}$$

The first two terms, with $C_s(x,y), C_t(x,y) \in \mathbb{R}$, are the cost of assigning the point (x,y) to the background and foreground, respectively, and the third term is the length of the boundary of ω. The parameter g is either a non-negative constant or an edge detector for the image u_0. In the previous two chapters, we used the level set method and piecewise constant function to solve the above minimization problem numerically. But the methods in these chapters do not guarantee any convergence to a global minimum.

The problem (5.1) can instead be written with a characteristic function $u(x,y) \in \{0,1\}$ for the region ω. Then the length term becomes the total variation of u:

$$\begin{aligned}\min_{u(x,y)\in\{0,1\}} \int_{\Omega} (1-u(x,y))\,C_s(x,y)\,dxdy + \int_{\Omega} u(x,y)\,C_t(x,y)\,dxdy \\ + g\int_{\Omega} |\nabla u(x,y)|\,dxdy.\end{aligned} \tag{5.2}$$

By relaxing the binary constraint set of $u(x, y)$ to the convex set $u(x, y) \in [0, 1]$, it is shown in [18] that a convex minimization problem can be solved globally and leads to a sequence of global binary optimizers when thresholding the solution $u^*(x, y) \in [0, 1]$ at any value in $(0, 1]$.

Max-flow and min-cut are known as the fundamental pair of the dual optimization problem defined over a graph. It can be used for dividing a data set into two regions when the data is represented by a set of nodes and edges.

This chapter, which is based on [4] by Yuan, Bae, Tai and Boykov, solves the minimization problem by using a maximal flow and minimal cut model over the continuous domain Ω, where every spatial point in the domain is linked to a source and a sink term. Therefore, (5.2) combined with a convex set $u(x, y) \in [0, 1]$ is also called a continuous min-cut model.

5.2 Discrete Max-Flow and Min-Cut

In the following section we study how the discrete max-flow and min-cut are formulated.

A graph $\mathcal{G}$, given as a pair $(\mathcal{V}, \mathcal{E})$, consists of a set of vertices $\mathcal{V}$ and a set of directed edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The two distinct vertices, the source $\{s\}$ and the sink $\{t\}$, are included in the vertex set $\mathcal{V}$, and a cost $C(e) \geq 0$ is delegated to every edge $e \in \mathcal{E}$.

If we consider the vertex set $\mathcal{V}$ with nodes of a two-dimensional nested grid, then the edge set $\mathcal{E}$ consists of two types of edges:

- The spatial edges:
  $e_n = (r, q)$, which is the edge between the neighbouring nodes $r, q \in \mathcal{V} \setminus \{s, t\}$ on the grid.

- The terminal edges:
  $e_s = (s, r)$ and $e_t = (r, t)$, which are the edges linking the source s and the sink t to the grid node $r \in \mathcal{V} \setminus \{s, t\}$.

Maximal Flow

Let us consider each edge $e \in \mathcal{E}$ as a pipe, and restrict the maximal flow $p(e)$ by the cost $C(e)$, i.e. we obtain the constraint

$$0 \leq p(e) \leq C(e). \tag{5.3}$$

In other words, $C(e)$ is the maximal capacity allowed on the pipe $p(e)$. At each vertex, a flow conservation is necessary:

$$\sum_{\substack{r \in \mathcal{V} \\ (r,q) \in \mathcal{E}}} p(r,q) - \sum_{\substack{r \in \mathcal{V} \\ (q,r) \in \mathcal{E}}} p(q,r) = 0, \qquad \forall\, q \in \mathcal{V} \setminus \{s,t\}. \tag{5.4}$$

We want to find the largest amount of flow allowed from the source s to the sink t. This problem, called the maximum flow problem, can equivalently be stated by finding the total amount of flow on the source edges, i.e.

$$\max_{p} \sum_{\substack{r \in \mathcal{V} \\ (s,r) \in \mathcal{E}}} p(s,r), \tag{5.5}$$

subject to (5.3) and (5.4).

Let us now study the constraints in the max-flow problem:

Capacity of spatial flows p:

The directed spatial edges $(r,q) \in \mathcal{E}$ and $(q,r) \in \mathcal{E}$, for $r,q \in \mathcal{V}\setminus\{s,t\}$, have the spatial flows $p(r,q)$ and $p(q,r)$, respectively, where the flows are constrained by

$$0 \le p(r,q) \le C(r,q), \qquad 0 \le p(q,r) \le C(q,r).$$

For a more simplified notation, a single flow $\tilde{p}(r,q)$ can be defined for each edge pair (r,q) and (q,r), and this flow can also be negative:

$$\tilde{p}(r,q) \;=\; p(r,q) - p(q,r),$$

with the constraint

$$-C(q,r) \le \tilde{p}(r,q) \le C(r,q).$$

Capacity of source flows p_s:

The edge $e_s(r) : s \to r$, where the terminal node s is linked to a node $r \in \mathcal{V} \setminus \{s,t\}$, has the source flow $p_s(r)$, which is directed from s to r. As we know that the flow has the capacity $C_s(r)$, the constraint of the flow is given by:

$$0 \le p_s(r) \le C_s(r).$$

Capacity of sink flows p_t:

The edge $e_t(r) : r \to t$, where a node $r \in \mathcal{V} \setminus \{s,t\}$ is linked to the terminal node t, has the source flow $p_t(r)$, which is directed from r to t. As we know that the flow has the capacity $C_t(r)$, the constraint of the flow is given by:

$$0 \le p_t(r) \le C_t(r).$$

Conservation of flows:

Let $\mathcal{N}(r)$ be the set of neighbouring nodes of $r \in \mathcal{V} \setminus \{s,t\}$. The flows, which are going into the node r, should be balanced by the flows going out from r. These flows include the spatial flows $p(r,q)$, with $q \in \mathcal{N}(r)$, the source flow $p_s(r)$ and the sink flow $p_t(r)$, and have the constraint

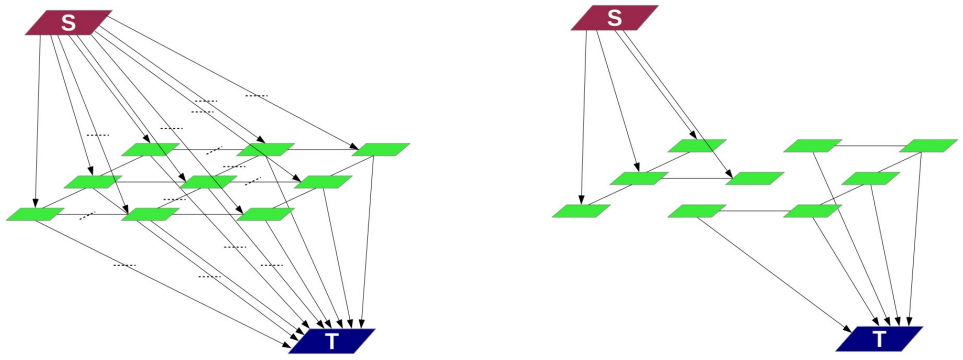
$$\left(\sum_{q \in \mathcal{N}(r)} \tilde{p}(q,r) \right) - p_s(r) + p_t(r) = 0.$$

Minimum Cut

Let the vertex set $\mathcal{V}_s$ consists of the source $\{s\}$ and let the vertex set $\mathcal{V}_t$ consists of the source $\{t\}$, such that

$$\mathcal{V} = \mathcal{V}_s \cup \mathcal{V}_t.$$

When the graph $\mathcal{G}$ is separated into two disjoint groups $\mathcal{V}_s$ and $\mathcal{V}_t$, i.e. $\mathcal{V}_s \cap \mathcal{V}_t = \emptyset$, then the separation is called a cut. There exists a path between s and the vertex $r \in \mathcal{V}_s$, and there exists a path between the vertex $q \in \mathcal{V}_t$ and t. The illustration in fig. 5.1 shows an example of a cut on a two-dimensional nested grid.



(a) A two-dimensional graph with a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$.

(b) The graph is cut into two disjoint groups $\mathcal{V}_s$ and $\mathcal{V}_t$.

Figure 5.1: A discrete max-flow and min-cut. In **(a)** some edges are marked with a dashed line. These edges will be cut/removed and the grid will be divided into two regions. The result is shown in **(b)**.

The edges which have a tail in $\mathcal{V}_s$ and a head in $\mathcal{V}_t$ are the edges which are cut. Therefore, we define the cost of this cut by the sum of the weighted

edges:

$$C\left(\mathcal{V}_s, \mathcal{V}_t\right) = \sum_{\substack{(r,q)\in\mathcal{E} \\ r\in\mathcal{V}_s\,,\, q\in\mathcal{V}_t}} C(r,q). \tag{5.6}$$

Finding the cut $(\mathcal{V}_s, \mathcal{V}_t)$, which gives the minimum cost $C(\mathcal{V}_s, \mathcal{V}_t)$, is called the minimum cut problem.

Edges with flow $p(e)$, for $e \in \mathcal{E}$, which attain the corresponding capacity $C(e)$ in the max-flow problem, i.e $p(e) = C(e)$, are the same edges as the edges with tail in $\mathcal{V}_s$ and head in $\mathcal{V}_t$ in the min-cut problem. It is proven in [19] that the max-flow problem (5.5) is equivalent to the min-cut problem (5.6).

5.3 Continuous Max-Flow and Min-Cut

In this subsection we construct a continuous max-flow and min-cut model by a direct generalization of the discrete max-flow and min-cut. These models are also called the primal and dual models, respectively.

5.3.1 Primal Model

Let Ω be a closed spatial two dimensional domain and let the source s and the sink t be the given terminals. For this spatial continuous setting, we consider for each point $(x, y) \in \Omega$:

- $p(x, y) \in C^{\infty}(\Omega)$: The spatial flow at (x, y).
- $p_s(x, y) \in \mathbb{R}$: The directed source flow from s to (x, y).
- $p_t(x, y) \in \mathbb{R}$: The directed sink flow from (x, y) to t.

The continuous max-flow model is achieved by considering the discrete max-flow with its constraints when the amount of grid nodes in $\mathcal{V}$ goes to infinity. Letting the flows $p(x, y)$, $p_s(x, y)$ and $p_t(x, y)$ be constrained by the capacities $C(x, y)$, $C_s(x, y)$ and $C_t(x, y)$, respectively, we obtain the following:

$$|p(x,y)| \le C(x,y), \qquad \forall (x,y) \in \Omega, \tag{5.7}$$
$$p_s(x,y) \le C_s(x,y), \qquad \forall (x,y) \in \Omega, \tag{5.8}$$
$$p_t(x,y) \le C_t(x,y), \qquad \forall (x,y) \in \Omega, \tag{5.9}$$
$$\nabla \cdot p(x,y) - p_s(x,y) + p_t(x,y) = 0, \qquad \text{a.e. for } (x,y) \in \Omega, \tag{5.10}$$
$$p(x,y) \cdot \vec{n} = 0, \qquad \forall (x,y) \in \partial\Omega, \tag{5.11}$$

where the notation a.e. is the short form of "almost everywhere" and $\vec{n}$ is the outward normal to the boundary $\partial\Omega$. Looking at (5.10),$\nabla \cdot p$ gives the total amount of spatial flows locally at the point (x, y), which is equivalent to the sum operator in the discrete flow conservation.

Comparing (5.8) and (5.9) to the discrete source and sink constraints, we see that the continuous flows p_s and p_t, and their corresponding capacities C_s and C_t, are no longer required to be non-negative. That is because the value of the directed flows imply the distribution from s to (x, y), or from (x, y) to t.

The continuous maximal flow model can now be expressed as

$$\sup_{p_s, p_t, p} \left\{ P\left(p_s,\, p_t,\, p\right) = \int_\Omega p_s(x, y)\, dxdy \right\} \tag{5.12}$$

subject to the constraints (5.7), (5.8), (5.9), (5.10), and (5.11). This model is called *the primal model* and the variables p_s, p_t, p are called the primal variables.

5.3.2 Primal-Dual model

In [4], it is shown how the continuous max-flow model (5.12) can be expressed as its corresponding *primal-dual model* by introducing the Lagrange multiplier function u, also known as the dual variable:

$$\inf_u \sup_{p_s, p_t, p} \left\{ E\left(p, p_s, p_t; u\right) = \int_\Omega p_s\, dxdy + \int_\Omega u(x, y)\left(\nabla \cdot p - p_s + p_t\right) dxdy \right\}$$

subject to

$$|p(x, y)| \leq C(x, y), \quad p_s(x, y) \leq C_s(x, y), \quad p_t(x, y) \leq C_t(x, y).$$

This primal-dual model can be rearranged, and is then given by

$$\inf_u \sup_{p_s, p_t, p} \left\{ E\left(p, p_s, p_t; u\right) = \int_\Omega \Big[(1 - u)\, p_s + u\, p_t + u\, \nabla \cdot p \Big] dxdy \right\} \tag{5.13}$$

subject to

$$|p(x, y)| \leq C(x, y), \quad p_s(x, y) \leq C_s(x, y), \quad p_t(x, y) \leq C_t(x, y).$$

The energy function $E(p, p_s, p_t; u)$ is linear in the primal functions p, p_s, p_t and in the dual function u; that is the constraints of the flows are convex. Therefore, from the minimax theorem [26], the energy function is convex u.s.c. for fixed primal functions and concave l.s.c. for fixed dual function. This implies that there exists at least one saddle point.

5.3.3 Dual Model

Let us first consider the maximization problem

$$f(q) \;=\; \sup_{p \leq C} \; q \cdot p, \tag{5.14}$$

for the scalars p, q and C. In order to maximize the value $p \cdot q$, we focus on the question how p is chosen for a given q:

- If $q < 0$, p needs to be negative infinity. This leads to $f(q) = +\infty$.
- If $q = 0$, $p \leq C$ is chosen and then $f(q) = 0$ becomes the maximum.
- If $q > 0$, we need to choose $p = C$ which results in $f(q) = C \cdot q$.

Using the conclusions above, $f(q)$ can be formulated as

$$f(q) \;=\; \begin{cases} q \cdot C\,, & \text{if} \quad q \geq 0, \\ \infty, & \text{if} \quad q < 0. \end{cases} \tag{5.15}$$

This can be applied to the primal-dual model (5.13) over the source flow p_s and the sink flow p_t.

We define

$$f_s(x,y) \;:=\; \sup_{p_s(x,y) \leq C_s(x,y)} (1 - u(x,y)) \cdot p_s(x,y),$$

and

$$f_t(x,y) \;:=\; \sup_{p_t(x,y) \leq C_t(x,y)} u(x,y) \cdot p_t(x,y).$$

From the conclusions above, we obtain the inequalities $1 - u(x,y) \geq 0$ and $u(x,y) \geq 0$, which leads to the constraint $0 \leq u(x,y) \leq 1$. If this constraint is not satisfied, the existence of at least one saddle point is contradicted because the primal-dual model becomes infinite. Now we obtain from (5.15) that

$$f_s(x,y) \;=\; (1 - u(x,y)) \cdot C_s(x,y)$$

and

$$f_t(x,y) \;=\; u(x,y) \cdot C_t(x,y).$$

The spatial flow $p(x,y)$ in (5.13) can be maximized by using the Divergence theorem and (5.11). For more details, see [20].

$$\begin{aligned} \sup_{|p(x,y)| \leq C(x,y)} \int_\Omega u \, \nabla \cdot p \, dxdy \;&=\; \sup_{|p(x,y)| \leq C(x,y)} \int_\Omega p \, |\nabla u| \, dxdy \\ &= \int_\Omega C \, |\nabla u| \, dxdy \end{aligned}$$

After maximizing the primal-dual model, it can be written as

$$\min_{u(x,y)\in[0,1]} \left\{ D(u) \;=\; \int_\Omega \Big[(1-u)\, C_s + u\, C_t + C\,|\nabla u| \Big]\, dxdy \right\}. \qquad (5.16)$$

This problem is the continuous min-cut model, also known as *the dual model*.

Summarizing all above, we see that the primal model (5.12), the primal-dual model (5.13), and the dual model (5.16) are equivalent to each other [4].

Global Binary Optimizers of the Min-Cut Problem

We begin this subsection by rewriting the dual model. For that purpose we assume that the function $C(x,y) \geq 0$ for all $(x,y) \in \Omega$ and that C is bounded and Borel measurable. The dual model can then be written as

$$\min_{u(x,y)\in[0,1]} \left\{ D(u) \;=\; \int_\Omega \Big[(1-u)\, C_s + u\, C_t + g\,|\nabla u| \Big]\, dxdy \right\} \qquad (5.17)$$

when $C(x,y) = g$ is constant $\forall (x,y) \in \Omega$. Nozawa showed in [22] that there is no duality gap between the primal and dual models with this type of spatial capacity functions. Since we want to segment an image, g can also be an edge detector which is zero at the edges and has large values elsewhere. This chapter uses the same edge detector (2.2) as in previous chapters. This edge detector satisfies the properties mentioned above.

Let us assume that the primal-dual pair p_s^*, p_t^*, p^* and u^* are the optimal values of the primal-dual model (5.13). Then we obtain that

$$P\,(p_s^*, p_t^*, p^*) \;=\; \int_\Omega p_s^*(x,y)\, dxdy, \qquad (5.18)$$

and

$$\nabla \cdot p^*(x,y) - p_s^*(x,y) + p_t^*(x,y) = 0, \qquad \text{a.e. for } (x,y) \in \Omega.$$

Now, let

$$\omega^l := \{\, (x,y)\,|\, u^*(x,y) \geq l,\, l \in (0,1] \,\}$$

be any level set of $u^*(x,y)$, and let its characteristic function be given by

$$u^l(x,y) \;=\; \begin{cases} 1, & u^*(x,y) \geq l, \\ 0, & u^*(x,y) < l. \end{cases}$$

Then, for $l \in (0, u^*(x,y)]$,

$$p_t^*(x,y) = C_t(x,y), \quad \forall\, (x,y) \in \omega^l,$$

which results in

$$p_s^*(x,y) = C_t(x,y) + \nabla \cdot p^*(x,y), \quad \text{a.e. for } (x,y) \in \omega^l.$$

In the same way, for $l \in (u^*(x,y), 1]$,

$$p_s^*(x,y) = C_s(x,y), \quad \forall\, (x,y) \in \Omega \setminus \omega^l.$$

By using the last two equations and thereafter applying the Divergence theorem, the total energy in (5.18) for the level set ω^l becomes

$$\begin{aligned} P\left(p_s^*, p_t^*, p^*\right) &= \int_{\Omega\setminus\omega^l} C_s(x,y)\, dxdy + \int_{\omega^l} \left[\, C_t(x,y) + \nabla \cdot p^*(x,y)\right] dxdy \\ &= \int_{\Omega\setminus\omega^l} C_s(x,y)\, dxdy + \int_{\omega^l} C_t(x,y)\, dxdy + \int_{\omega^l} \nabla \cdot p^*(x,y)\, dxdy \\ &= \int_{\Omega\setminus\omega^l} C_s(x,y)\, dxdy + \int_{\omega^l} C_t(x,y)\, dxdy + \int_{\partial\omega^l} p^*(x,y) \cdot \vec{n}\, dl, \end{aligned}$$

where $\vec{n}$ is the the outward normal to the boundary $\partial\omega^l$. Since we have that $p^*(x,y) \cdot \vec{n} = g$, for all $(x,y) \in \partial\omega^l \setminus (\partial\omega^l \cap \partial\Omega)$, we obtain that

$$P\left(p_s^*, p_t^*, p^*\right) = \int_{\Omega\setminus\omega^l} C_s(x,y)\, dxdy + \int_{\omega^l} C_t(x,y)\, dxdy + g\, |\partial\omega^l \setminus (\partial\omega^l \cap \partial\Omega)|.$$

We conclude that the binary function u^l, which is in the relaxed convex set $[0,1]$, solves the non-convex segmentation problem (5.1) globally. Hence, solving the continuous maximal flow problem (5.12) is an implicit result of segmenting the region Ω by solving the continuous minimal cut problem with the optimal multiplier $u^*(x,y)$.

5.4 Continuous Max-Flow Based Algorithm

In this section, as done in [4], we solve the primal-dual problem (5.13) by introducing the augmented Lagrangian method, see [23], for this problem:

$$\begin{aligned} \mathcal{L}_c\left(p_s,\, p_t,\, p,\, u\right) := &\int_\Omega p_s\, dxdy + \int_\Omega u\left(\nabla \cdot p - p_s + p_t\right) dxdy \\ &- \frac{c}{2}\left\|\nabla \cdot p - p_s + p_t\right\|^2, \end{aligned}$$

where $\|f(x,y)\|^2 = \int_\Omega f(x,y)^2\,dxdy$ and $c > 0$ is a scalar. Now, we optimize the variables one by one in $L_c(p_s, p_t, p, u)$ by keeping the other variables fixed at every iteration, but u is updated by the current value of u and the optimized variables p_s, p_t, p. We assume that spatial discretization can be applied to the divergence, gradient and integration operators $(\nabla\cdot, \nabla, \int)$ and the flow variables p, p_s, p_t. So, when the grid size goes to zero, the discrete gradient and divergence terms converge to the respective continuous terms. Then the duality gap goes to zero.

The next step is to show how every variable, at the kth iteration, is optimized.

Optimizing w.r.t. p:

We can optimize p by keeping p_s, p_t and u fixed at the kth iteration:

$$
\begin{aligned}
p^{k+1} :=& \underset{\|p\|_\infty \le g}{\arg\max}\ \mathcal{L}_c(p_s^k, p_t^k, p, u^k)\\
=& \int_\Omega \Big[u^k\big(\nabla\cdot p - p_s^k + p_t^k\big) - \frac{c}{2}\big(\nabla\cdot p - p_s^k + p_t^k\big)^2 \Big]\,dxdy\\
=& \int_\Omega \Big(-\frac{c}{2}\Big[\big(\nabla\cdot p - p_s^k + p_t^k\big)^2 - \frac{2\,u^k}{c}\big(\nabla\cdot p - p_s^k + p_t^k\big) \Big] \Big)\,dxdy\\
=& -\frac{c}{2}\int_\Omega \Big[\big(\nabla\cdot p - p_s^k + p_t^k\big)^2 - \frac{2\,u^k}{c}\big(\nabla\cdot p - p_s^k + p_t^k\big)\\
&+ \Big(\frac{u^k}{c}\Big)^2 - \Big(\frac{u^k}{c}\Big)^2 \Big]\,dxdy\\
=& -\frac{c}{2}\int_\Omega \Big(\Big[\,\nabla\cdot p - p_s^k + p_t^k - \frac{u^k}{c}\,\Big]^2 - \Big(\frac{u^k}{c}\Big)^2 \Big)\,dxdy\\
=& \underset{\|p\|_\infty \le g}{\arg\max}\ -\frac{c}{2}\int_\Omega \Big[\,\nabla\cdot p - p_s^k + p_t^k - \frac{u^k}{c}\,\Big]^2\,dxdy\\
=& \underset{\|p\|_\infty \le g}{\arg\max}\ -\frac{c}{2}\int_\Omega \Big[\,\nabla\cdot p - F^k\,\Big]^2\,dxdy\\
=& \underset{\|p\|_\infty \le g}{\arg\max}\ -\frac{c}{2}\|\,\nabla\cdot p - F^k\,\|^2,
\end{aligned}
$$

where $F^k = p_s^k - p_t^k + \dfrac{u^k}{c}$.

Let

$$M(p) := \int_\Omega \frac{c}{2}\Big[\nabla\cdot p - F^k\Big]^2\,dxdy,$$

then we use the Gâteaux differential [26, p. 23] of the functional $M(p)$ with respect to the function p in the direction μ, and apply the Divergence theorem, to obtain that

$$\begin{aligned} D_\mu M(p) &= \lim_{\tau \to 0} \frac{1}{\tau} \left[M(p + \tau\mu) - M(p) \right] \\ &= c \int_\Omega \left[\nabla \cdot p - F^k \right] \nabla \cdot \mu \, dxdy \\ &= -c \int_\Omega \nabla \left[\nabla \cdot p - F^k \right] \mu \, dxdy + c \int_{\partial\Omega} \mu \left[\nabla \cdot p - F^k \right] \cdot \vec{n} \, ds, \end{aligned}$$

where $\vec{n}$ is the outward normal to the boundary $\partial\Omega$.

We require that $[\nabla \cdot p - F^k] \cdot \vec{n} = 0$ on $\partial\Omega$, and get

$$\begin{aligned} D_\mu M(p) &= -c \int_\Omega \nabla \left(\nabla \cdot p - F^k \right) \mu \, dxdy \\ \Rightarrow \quad \frac{\partial M}{\partial p} &= -c \nabla \left(\nabla \cdot p - F^k \right). \end{aligned}$$

Now, one gradient descent step with the step size γ is applied, and thereafter we use the Euclidean projection $\mathbf{\Pi}_g$ onto the convex set $C_g = \left\{ q \,\middle|\, \|q\|_\infty \leq g \right\}$. Therefore,

$$p^{k+1} = \mathbf{\Pi}_g \Big[p^k + \gamma \, c \nabla \left(\nabla \cdot p^k - F^k \right) \Big],$$

where $\gamma > 0$ is a small step size.

Optimizing w.r.t. p_s:

Again, the optimization of p_s is done by keeping p, p_t and u fixed at the kth iteration, such that

$$\begin{aligned} p_s^{k+1} &:= \underset{p_s \leq C_s}{\arg\max} \; \mathcal{L}_c \left(p_s, \, p_t^k, \, p^{k+1}, \, u^k \right) \\ &= \int_\Omega p_s \, dxdy + \int_\Omega \Big[u^k \left(\nabla \cdot p^{k+1} - p_s + p_t^k \right) - \frac{c}{2} \left(\nabla \cdot p^{k+1} - p_s + p_t^k \right)^2 \Big] \, dxdy \\ &= \int_\Omega p_s \, dxdy - \frac{c}{2} \int_\Omega \Big[\left(\nabla \cdot p^{k+1} - p_s + p_t^k \right)^2 - \frac{2u^k}{c} \left(\nabla \cdot p^{k+1} - p_s + p_t^k \right) \Big] dxdy \\ &= \int_\Omega p_s \, dxdy - \frac{c}{2} \int_\Omega \Big[\left(\nabla \cdot p^{k+1} - p_s + p_t^k \right)^2 - \frac{2\,u^k}{c} \left(\nabla \cdot p^{k+1} - p_s + p_t^k \right) \\ &\qquad + \left(\frac{u^k}{c} \right)^2 - \left(\frac{u^k}{c} \right)^2 \Big] \, dxdy \end{aligned}$$

$$
\begin{aligned}
&= \int_\Omega p_s \, dxdy - \frac{c}{2} \int_\Omega \Big[\, (\nabla \cdot p^{k+1} - p_s + p_t^k - \frac{u^k}{c}\,)^2 - \Big(\frac{u^k}{c}\Big)^2 \Big] \, dxdy \\
&= \arg\max_{p_s \le C_s} \left(\int_\Omega p_s \, dxdy - \frac{c}{2} \int_\Omega \Big[\nabla \cdot p^{k+1} - p_s + p_t^k - \frac{u^k}{c} \Big]^2 dxdy \right) \\
&= \arg\max_{p_s \le C_s} \left(\int_\Omega p_s \, dxdy - \frac{c}{2} \int_\Omega \Big[\, p_s \; - \; G^k \Big]^2 dxdy \right) \\
&= \arg\max_{p_s \le C_s} \left(\int_\Omega p_s \, dxdy - \frac{c}{2} \, \| \, p_s \; - \; G^k \, \|^2 \right),
\end{aligned}
$$

where $G^k = \nabla \cdot p^{k+1} + p_t^k - \dfrac{u^k}{c}$.

Let

$$
M_s(p_s) := \int_\Omega \Big[\, p_s - \frac{c}{2} \, \big(\, p_s \; - \; G^k \big)^2 \Big] \, dxdy,
$$

then the optimized $p_s(x, y)$ can be computed pointwise at each $(x, y) \in \Omega$ by the Gâteaux differential of the functional $M_s(p_s)$ with respect to the function p_s in the direction μ, such that

$$
\begin{aligned}
D_\mu M_s(p_s) \; &= \lim_{\tau \to 0} \frac{1}{\tau} \big[\, M_s(p_s + \tau\mu) - M_s(p_s) \, \big] \\
&= \int_\Omega \big[1 - c \, \big(p_s - G^k \big) \big] \, \mu \; dxdy,
\end{aligned}
$$

$$
\Rightarrow \quad \frac{\partial \, M_s}{\partial \, p_s} \; = \; 1 - c \, \big(p_s - G^k \big) \; = \; 0, \qquad\qquad \Rightarrow \quad p_s \; = \; G^k + \frac{1}{c}.
$$

Since we also have the constraint $p_s \le C_s$, we obtain that

$$
p_s^{k+1}(x, y) = \min \left(\, G^k(x, y) + \frac{1}{c} \, , \, C_s(x, y) \, \right).
$$

Optimizing w.r.t. p_t:

As before, we optimize p_t by keeping p, p_s and u fixed at the kth iteration:

$$
\begin{aligned}
p_t^{k+1} &:= \arg\max_{p_t \le C_t} \ \mathcal{L}_c\left(p_s^{k+1},\, p_t,\, p^{k+1},\, u^k\right) \\
&= \int_\Omega \Big[\, u^k \left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right) - \frac{c}{2}\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right)^2 \Big]\, dxdy \\
&= -\frac{c}{2}\int_\Omega \left[\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right)^2 - \frac{2\,u^k}{c}\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right)\right] dxdy \\
&= -\frac{c}{2}\int_\Omega \left[\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right)^2 - \frac{2\,u^k}{c}\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t\right)\right. \\
&\qquad\qquad \left. + \left(\frac{u^k}{c}\right)^2 - \left(\frac{u^k}{c}\right)^2\right] dxdy \\
&= -\frac{c}{2}\int_\Omega \left[\left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t - \frac{u^k}{c}\right)^2 - \left(\frac{u^k}{c}\right)^2\right] dxdy \\
&= \arg\max_{p_t \le C_t} -\frac{c}{2}\int_\Omega \left(\nabla \cdot p^{k+1} - p_s^{k+1} + p_t - \frac{u^k}{c}\right)^2 dxdy \\
&= \arg\max_{p_t \le C_t} -\frac{c}{2}\int_\Omega \left(p_t - H^k\right)^2\, dxdy \\
&= \arg\max_{p_t \le C_t} -\frac{c}{2}\,\| p_t - H^k \|^2,
\end{aligned}
$$

where $H^k = -\nabla \cdot p^{k+1} + p_s^{k+1} + \dfrac{u^k}{c}$.

Let

$$M_t(p_t) := -\frac{c}{2}\int_\Omega \left(p_t - H^k\right)\, dxdy,$$

then, similarly as for the former variable, the optimized $p_t(x, y)$ can be computed pointwise at each $(x, y) \in \Omega$ by the Gâteaux differential of the functional $M_t(p_t)$ with respect to the function p_t in the direction μ, such that

$$
\begin{aligned}
D_\mu M_t(p_s) &= \lim_{\tau \to 0} \frac{1}{\tau}\big[\, M_t(p_t + \tau\mu) - M_t(p_t)\,\big] \\
&= \int_\Omega \big[\, -c\left(p_t - H^k\right)\big]\, \mu\, dxdy,
\end{aligned}
$$

$$
\Rightarrow \quad \frac{\partial\, M_t}{\partial\, p_t} = -c\left(p_s - H^k\right) = 0, \qquad \Rightarrow \quad p_t = H^k.
$$

Since we also have the constraint $p_t \le C_t$, we obtain that

$$p_t^{k+1}(x, y) = \min\left(\, H^k(x, y)\,,\, C_t(x, y)\,\right).$$

Updating u:

We can update u by using the method of multipliers:

$$u^{k+1} \;=\; u^k - c\left(\nabla\cdot p^{k+1} - p_s^{k+1} + p_t^{k+1}\right).$$

Algorithm 3 Max-flow Min-cut algorithm

Initialize: $p^0 = 0$, $p_s^0 = \min(C_s - C_t)$, $p_t^0 = \min(C_s - C_t)$,

$$u^0 = \begin{cases} 1, & \text{if } (C_s - C_t) \geq 0, \\ 0, & \text{if } (C_s - C_t) < 0, \end{cases}$$

and set $k = 0$:

repeat

- Optimize p by keeping p_s, p_t and u fixed:

$$p^{k+1} \;\leftarrow \mathbf{\Pi}_g\Big[p^k \;+\; \gamma\, c\nabla\left(\nabla\cdot p^k - F^k\right)\Big],$$

  where $F^k = p_s^k - p_t^k + \dfrac{u^k}{c}$.

- Optimize p_s by keeping p, p_t and u fixed:

$$p_s^{k+1} \;\leftarrow\; \min\left(G^k(x) + \frac{1}{c}\,,\, C_s(x)\right),$$

  where $G^k = \nabla\cdot p^{k+1} + p_t^k - \dfrac{u^k}{c}$.

- Optimize p_t by keeping p, p_s and u fixed:

$$p_t^{k+1} \;\leftarrow\; \min\left(H^k(x)\,,\, C_t(x)\right),$$

  where $H^k = -\nabla\cdot p^{k+1} + p_s^{k+1} + \dfrac{u^k}{c}$.

- Update u:

$$u^{k+1} \;\leftarrow\; u^k - c\left(\nabla\cdot p^{k+1} - p_s^{k+1} + p_t^{k+1}\right).$$

- $k \leftarrow k+1$ if the *until*-condition is not satisfied.

until $\dfrac{\|u^{k+1} - u^k\|}{\|u^{k+1}\|} < err.$

5.5 Experimental Results

The code for the Fast CMF Algorithm is written by J. Yuan. After incorporating the edge detector (2.2) into this code, we obtain the experimental results. In the numerical experiments we first have to choose two grey values u_s and u_t similar to the foreground and the background colour, respectively, such that we can build up the data terms:

$$C_s = |u_0(x,y) - u_s|^2, \qquad C_t = |u_0(x,y) - u_t|^2.$$

The fixed parameters in the experiments are $c = 1$, $err = 10^{-6}$, and $\gamma = 0.16$. When the parameter g is an edge detector, σ, a and b in (2.2) have to be chosen for each image. If g is not an edge detector, this parameter is interpreted as a penalty parameter to the total variation term.

The segmentation result for all images below is optimal when g is the edge detector (2.2). Furthermore, it gives more detailed segmentation. For example, in fig. 5.7e we see that only the jasmin flowers are segmented, while in fig. 5.7c some of the background is included in the segmented result. This also happens in fig. 5.12 when the squirrel image is being segmented.

Some of the results below can be compared with the results from previous chapters. We see that using the model in this chapter gives much better segmentation of the images.

(a) The original image.

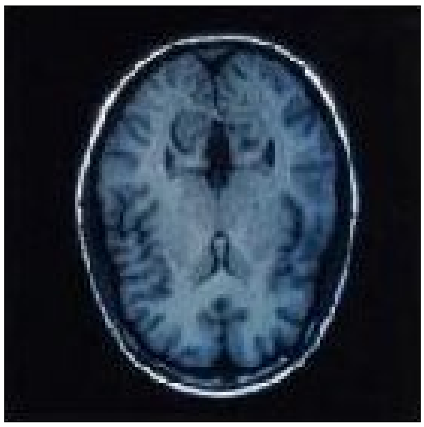
(b) Segmented result without edge detector with $g = 0.5$.

(c) Segmented result with the original image.

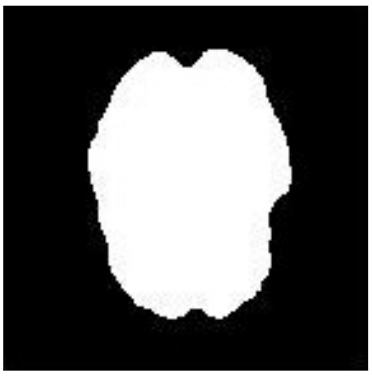
(d) Segmented result with edge detector (2.2) using $\sigma = 5$, $a = 0.5$, $b = 0.5$.

(e) Segmented result with the original image.

Figure 5.2: Segmentation of an owl image without and with edge detector.



(a) The original image.



(b) Segmented result without edge detector with $g = 1$.

(c) Segmented result with the original image.

(d) Segmented result with edge detector (2.2) using $\sigma = 0.8$, $a = 1$, $b = 80$.

(e) Segmented result with the original image.

Figure 5.3: Segmentation of a brain scan image without and with edge detector.

(a) The original image.

(b) Segmented result without edge detector with $g = 0.5$.

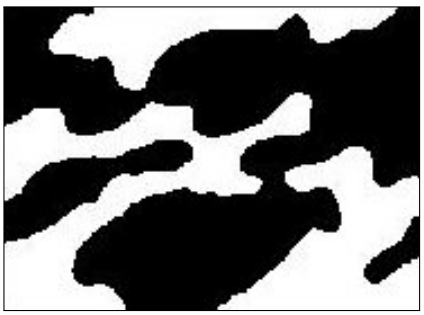
(c) Segmented result with the original image.

(d) Segmented result with edge detector (2.2) using $\sigma = 5$, $a = 0.5$, $b = 1$.

(e) Segmented result with the original image.

Figure 5.4: Segmentation of a night-light image over Europe without and with edge detector.

(a) The original image.



(b) Segmented result without edge detector with $g = 0.5$.

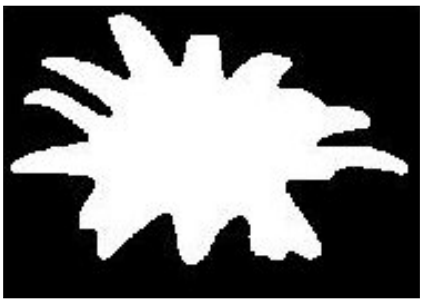
(c) Segmented result with the original image.

(d) Segmented result with edge detector (2.2) using $\sigma = 0.5$, $a = 0.5$, $b = 90$.

(e) Segmented result with the original image.

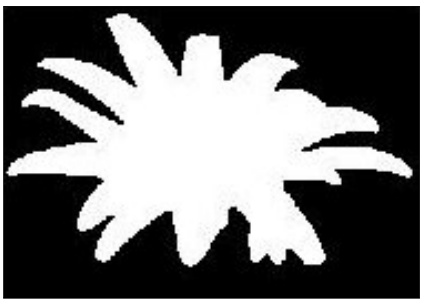
Figure 5.5: Segmentation of a fish shoal image without and with edge detector.

(a) The original image.



(b) Segmented result without edge detector with $g = 0.5$.

(c) Segmented result with the original image.



(d) Segmented result with edge detector (2.2) using $\sigma = 2$, $a = 0.5$, $b = 10$.

(e) Segmented result with the original image.

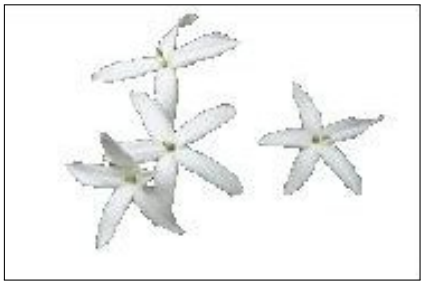
Figure 5.6: Segmentation of a flower image without and with edge detector.

(a) The original image.

(b) Segmented result without edge detector with $g = 1$.

(c) Segmented result with the original image.

(d) Segmented result with edge detector (2.2) using $\sigma = 0.8$, $a = 1$, $b = 30$.



(e) Segmented result with the original image.

Figure 5.7: Segmentation of an image of Jasmin flowers without and with edge detector.

(a) The original image.

(b) Segmented result without edge detector with $g = 1$.

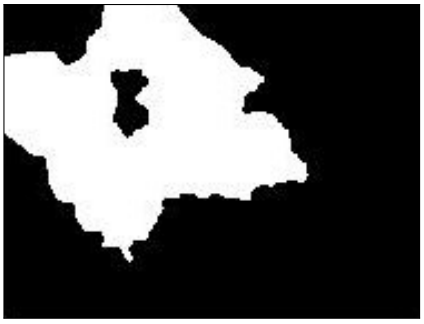
(c) Segmented result with the original image.

(d) Segmented result with edge detector (2.2) using $\sigma = 1$, $a = 1$, $b = 40$.

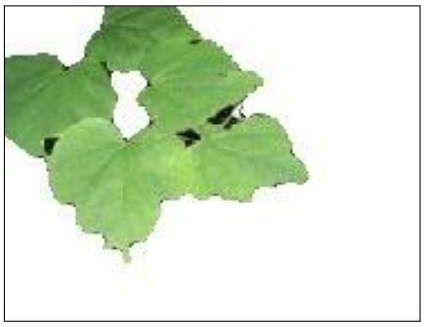
(e) Segmented result with the original image.

Figure 5.8: Segmentation of an image of a set of keys without and with edge detector.

(a) The original image.

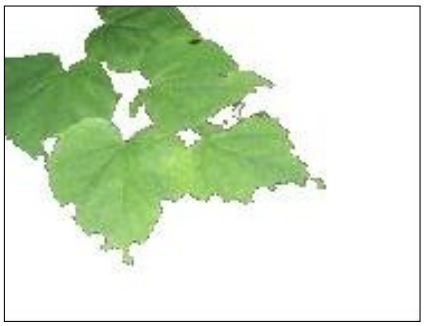


(b) Segmented result without edge detector with $g = 0.5$.



(c) Segmented result with the original image.

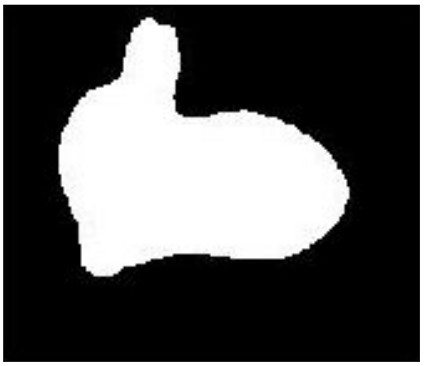
(d) Segmented result with edge detector (2.2) using $\sigma = 0.5$, $a = 0.5$, $b = 70$.



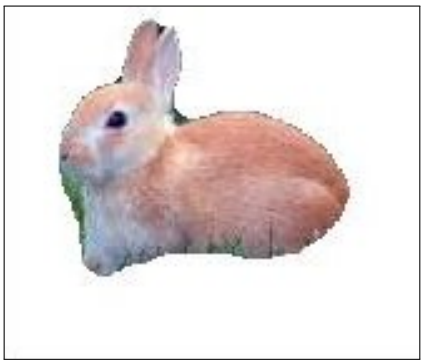
(e) Segmented result with the original image.

Figure 5.9: Segmentation of an image of leaves without and with edge detector.

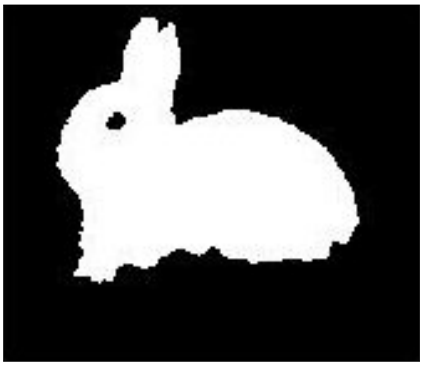
(a) The original image.



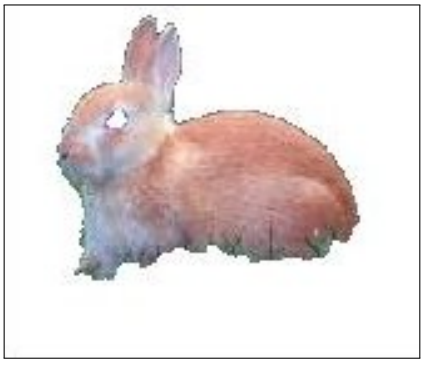
(b) Segmented result without edge detector with $g = 1$.



(c) Segmented result with the original image.



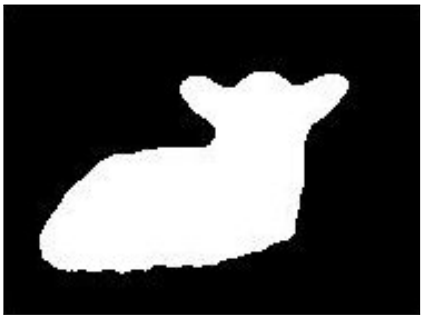
(d) Segmented result with edge detector (2.2) using $\sigma = 2$, $a = 1$, $b = 20$.



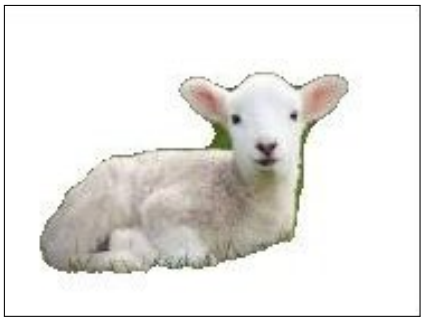
(e) Segmented result with the original image.

Figure 5.10: Segmentation of a rabbit image without and with edge detector.

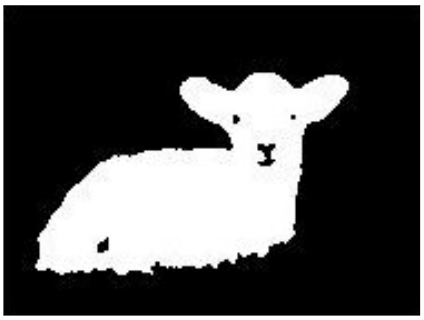
(a) The original image.



(b) Segmented result without edge detector with $g = 0.7$.



(c) Segmented result with the original image.

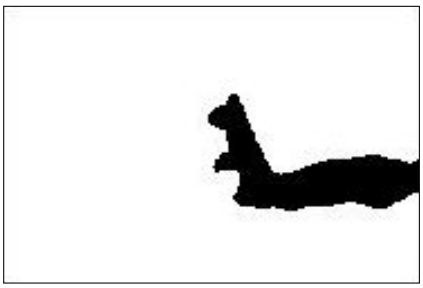


(d) Segmented result with edge detector (2.2) using $\sigma = 0.8$, $a = 0.7$, $b = 65$.

(e) Segmented result with the original image.

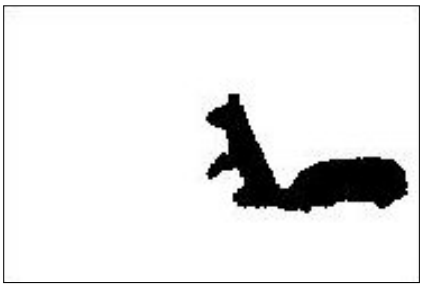
Figure 5.11: Segmentation of a sheep image without and with edge detector.

(a) The original image.



(b) Segmented result without edge detector with $g = 0.5$.

(c) Segmented result with the original image.



(d) Segmented result with edge detector (2.2) using $\sigma = 5$, $a = 0.5$, $b = 0.5$.

(e) Segmented result with the original image.

Figure 5.12: Segmentation of a squirrel image without and with edge detector.

Chapter 6

Summary and Conclusion

In this master thesis, we focused on three different variational energy minimization problems which arise in image segmentation.

The first minimization problem is an active contour model based on the level set method and Mumford-Shah segmentation. Minimizing the energy functional leads to an Euler-Lagrange equation, which is afterwards numerically solved. The experimental results show that by including an experimental edge detector, we obtain a better segmentation.

The second problem uses a piecewise constant level set method also applied to Mumford-Shah segmentation. The approximation is such that the level set function is discontinuous when it converges. Since the energy functional is smooth, a simple gradient method is used to solve the minimization problem. The experimental results indicate that this method gives a good segmentation. In this case, the CFL stability condition of the numerical scheme has to be taken into account. By comparing this model with the previous model, we see that the second model gives a more accurate segmentation result. It is important to remark that both models are non-convex, which may give many local minima.

In the third minimization method, we use the continuous max-flow min-cut model to segment images. Employing variational techniques, we show the duality between the max-flow min-cut model in the spatially continuous setting. Since this method is convex, the continuous max-flow based algorithm provides a fast and stable numerical scheme. For more details, see [4]. In the experimental results, we see that using an edge detector with the continuous max-flow min-cut model gives a much better and accurate segmentation compared to the first two models.

Bibliography

Books and Articles

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2001.

[2] T. F. Chan and L. A. Vese, "Active contours without edges," *Trans. Img. Proc.*, vol. 10, pp. 266–277, Feb. 2001.

[3] J. Lie, M. Lysaker, and X.-C. Tai, "A binary level set model and some applications to mumford-shah image segmentation.," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1171–1181, 2006.

[4] J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov, "A spatially continuous max-flow and min-cut framework for binary labeling problems," *Numer. Math.*, vol. 126, no. 3, pp. 559–587, 2014.

[5] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[6] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.

[7] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, no. 1, pp. 179–195, 1996.

[8] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, 1995.

[9] L. C. Evans, *Partial differential equations*, vol. 19 of *Graduate Studies in Mathematics*. Providence, RI: American Mathematical Society, second ed., 2010.

[10] X.-C. Tai and T. F. Chan, "A survey on multiple level set methods with applications for identifying piecewise constant functions," *Int. J. Numer. Anal. Model.*, vol. 1, no. 1, pp. 25–47, 2004.

[11] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer Series in Operations Research and Financial Engineering, New York: Springer, second ed., 2006.

[12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *INTERNATIONAL JOURNAL OF COMPUTER VISION*, vol. 1, no. 4, pp. 321–331, 1988.

[13] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numer. Math.*, vol. 66, no. 1, pp. 1–31, 1993.

[14] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vision*, vol. 22, pp. 61–79, Feb. 1997.

[15] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Topology-independent shape modeling scheme," *Proc. SPIE*, vol. 2031, pp. 246–258, 1993.

[16] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *Int. J. Comput. Vision*, vol. 50, pp. 271–293, Dec. 2002.

[17] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics, New York: Academic Press Inc. [Harcourt Brace Jovanovich Publishers], 1982.

[18] T. F. Chan, S. Esedoḡlu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2006.

[19] R. J. Vanderbei, *Linear programming*. International Series in Operations Research & Management Science, 196, New York: Springer, fourth ed., 2014. Foundations and extensions.

[20] E. Giusti, *Minimal surfaces and functions of bounded variation*. Department of Pure Mathematics, Australian National University, Canberra, 1977. With notes by Graham H. Williams, Notes on Pure Mathematics, 10.

[21] T. F. Chan and J. Shen, *Image processing and analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. Variational, PDE, wavelet, and stochastic methods.

[22] R. Nozawa, "Examples of max-flow and min-cut problems with duality gaps in continuous networks," *Math. Programming*, vol. 63, no. 2, Ser. A, pp. 213–234, 1994.

[23] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd ed., 1999.

[24] J. M. Hyman and M. Shashkov, "Adjoint operators for the natural discretizations of the divergence, gradient, and curl on logically rectangular grids," *APPL. NUMER. MATH*, vol. 25, pp. 413–442, 1997.

[25] J. M. Hyman and M. Shashkov, "Natural discretizations for the divergence, gradient, and curl on logically rectangular grids," *Computers & Mathematics with Applications*, vol. 33, no. 4, pp. 81 – 104, 1997.

[26] I. Ekeland and R. Témam, *Convex analysis and variational problems*, vol. 28 of *Classics in Applied Mathematics*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), english ed., 1999. Translated from the French.

[27] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, vol. 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[28] G. Zheng and L.-P. Nolte, "Surface reconstruction of bone from x-ray images and point distribution model incorporating a novel method for 2d-3d correspondence," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2237–2244, 2006.

Images

[29] 1hdwallpapers, "The snowy owl in flight." `http://1hdwallpapers.com/the_snowy_owl_in_flight-wallpaper.html`, 2013.

[30] K. Than, "Brain reading." National Geographic News, `http://news.nationalgeographic.com/news/bigphotos/47438275.html`, 2008.

[31] J. Llamas, "Europe night lights." Futuro Sostenible, `http://futurosostenible.greenglobe.es/?tag=contaminacion-luminica-2`, 2011.

[32] P. Harvey and P. Harvey, "Large shoals of fish." Pat and Paul go travelling, http://patandpaulharvey.blogspot.no/2009/06/diving-on-barrier-reef.html, 2009.

[33] FoeNyx, "Flower of a hybrid of hatiora x graeseri." Pat and Paul go travelling, http://en.wikibooks.org/wiki/Template:Flowers, 2005.

[34] F. F. Fotos, "The sweet-scented jasmin." Floral Friday Fotos, http://floralfridayfoto.blogspot.no/2012/05/fff25-sweet-scented-jasmine.html, 2012.

[35] S. Shephard, "We trust you." Graph Paper Press, http://www.scottshephard.com/2010/08/25/we-trust-you/, 2010.

[36] D. Sheriff, "Leaves." The Citrus Guy, http://thecitrusguy.blogspot.no/2013_07_01_archive.html, 2013.

[37] DifferenceBetween.info, "Rabbit." http://www.differencebetween.info/difference-between-rabbit-and-bunny, 2012-2014.

[38] thickethouse, "The lamb has left already." Thoughts from Thicket House, http://thickethouse.wordpress.com/2014/03/03/the-lamb-has-left-already/, 2012-2014.

[39] P. Kratochvil, "Black squirrel." http://www.publicdomainpictures.net/view-image.php?image=10693, 2007-2014.