

QUEENS'S UNIVERSITY

MTHE 493 THESIS REPORT

GROUP 9 PROJECT B-2

Tracking Stem Cells in an Image Sequence

Authors:

Declan Colwell (10173388)
Imogen den Otter Moore (20013012)
Laura Di Luch (20008194)
Peter Kyle (10197489)
Stefan Robb (20027476)

Supervisor: Dr. Abdol-Reza Mansouri

April, 2020

We are very grateful for the guidance and support of Dr. Abdol-Reza Mansouri throughout this project.

Abstract

Automated region tracking is the tracking of a selected region of interest across an image sequence. The objective of this paper was to design, implement, and test a region tracking algorithm that can be used to track stem cells through image sequences. The social, economic, environmental and ethical implications of using such an algorithm in a medical context were also explored. This paper approached region tracking using a calculus of variations approach, which involved re-framing the task as a minimization problem. A functional was designed that is a linear combination of five functionals that each represent properties of the region of interest that were assumed to remain invariant between frames. These invariant properties were intensity, mean intensity, intensity variance, area and shape. The functional was designed such that its unique global minimizer is the region being tracked. The algorithm was implemented using Python, and its tracking accuracy was evaluated using the Jaccard similarity index – a metric that represents the fraction of pixels in the predicted region that match the ground truth. The target Jaccard similarity index for the Cell Tracking Challenge data set was 0.914, which is the average accuracy of the top three algorithms submitted to the Cell Tracking Challenge for the considered dataset. The target Jaccard similarity index for the NYU Center for Genomics and Systems Biology data set was 0.95, established in a paper by Cicconet et al. The target value was not achieved for the first test data set, but was achieved for the second test data set. This indicates that the designed algorithm has the potential to be very accurate, but is not consistently so. Future work would involve building on the functional to ensure it fully captures all invariant properties, and working to make the algorithm more efficient so as to decrease its run-time.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Existing Work | 1 |
| 1.3 | Introduction to Cell Tracking | 1 |
| 2 | Problem Description | 4 |
| 2.1 | Problem Formulation | 4 |
| 2.2 | Problem Statement | 4 |
| 2.3 | An Overview of Stem Cell Research | 4 |
| 3 | Mathematical Background | 6 |
| 3.1 | Introduction | 6 |
| 3.2 | Notation | 6 |
| 3.3 | Calculus of Variations | 7 |
| 3.4 | Gradient Descent | 10 |
| 3.4.1 | Minimizing Contour Integrals | 12 |
| 3.4.2 | Minimizing Surface Integrals | 14 |
| 3.5 | Level Set Method | 16 |
| 3.5.1 | Curvature in the Level Set Method | 17 |
| 3.6 | Discretization | 18 |
| 3.7 | Validating Design Approach | 21 |
| 4 | Design Setup | 23 |
| 4.1 | Choice of Programming Language | 23 |
| 4.2 | Pseudocode | 23 |
| 4.3 | Algorithm Time Step | 24 |
| 4.4 | Stopping Condition | 24 |
| 4.5 | Image Definition | 25 |
| 4.6 | Functional Evaluation Metric | 25 |
| 5 | Design Solution | 27 |
| 5.1 | Invariant Properties | 27 |
| 5.2 | Functional Design | 27 |
| 5.2.1 | Length Functional | 27 |
| 5.2.2 | Intensity Functional | 28 |
| 5.2.3 | Mean Intensity Functional | 28 |
| 5.2.4 | Area Functional | 29 |
| 5.2.5 | Intensity Variance Functional | 30 |
| 5.3 | Region Tracking Functional | 31 |
| 6 | Implementation and Evaluation of Results | 33 |
| 7 | Societal, Ethical, Environmental and Economic Impacts | 40 |
| 7.1 | Societal Impact | 40 |
| 7.2 | Ethics | 40 |
| 7.3 | Environmental Impact | 40 |
| 7.4 | Economic Impact | 41 |
| 8 | Conclusions and Future Work | 43 |

| | |
|-----------------------------|-----------|
| A Important Theorems | 44 |
| References | 45 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Initial image of 4 stem cells [1] | 2 |
| 1.2 | Image of the cells a few seconds later, beginning to multiply [1] | 2 |
| 1.3 | Image of the cells a few more seconds later, continuing multiplying [1] | 2 |
| 2.1 | Healthy brain compared to the brain of a patient with Alzheimer's Disease [6] | 5 |
| 3.1 | Visualization of gradient descent | 11 |
| 3.2 | Propagating curve | 14 |
| 3.3 | Computational grid | 21 |
| 3.4 | Mean curvature flow validation | 22 |
| 4.1 | GFP-GOWT1 mouse stem cell baseline image set [21] | 25 |
| 5.1 | Testing the intensity functional | 28 |
| 5.2 | Testing mean intensity functional | 29 |
| 5.3 | Testing combined mean intensity and area functionals | 30 |
| 5.4 | Testing combined mean intensity and variance functionals | 31 |
| 6.1 | Tracking performance | 33 |
| 6.2 | Image filtering results | 33 |
| 6.3 | Effects of filtering on tracking performance | 34 |
| 6.4 | Tracking on mouse stem cell data set | 35 |
| 6.5 | Mouse embryo cell image frame | 36 |
| 6.6 | Tracking on mouse embryo cell data set | 37 |
| 6.7 | Dilation filtering results | 38 |
| 6.8 | Effects of dilation filtering on tracking performance | 38 |
| 6.9 | Adaptive thresholding filtering results | 39 |

List of Tables

| | | |
|-----|--|----|
| 3.1 | Forward and backward difference approximations | 19 |
| 3.2 | Central difference approximations | 19 |
| 3.3 | Discretization of the forward and backward difference approximations | 20 |
| 3.4 | Discretization of the central difference approximations | 20 |
| 6.1 | Results of filtering on the cell images | 34 |

1 Introduction

1.1 Motivation

Automated region tracking is both an important and challenging problem in the field of computer vision. The goal of computer vision is to have computers see, identify and analyse images – essentially automating what is done by the human visual system. The development of sophisticated and reliable region tracking algorithms can have useful applications in several areas including sports analysis and strategy, biomechanics (for example in marker-less joint tracking), facial recognition, and medical image processing. The process of region tracking begins with the selection of a region of interest in the initial frame of an image sequence. The task of a region tracking algorithm is to identify the region of interest in the subsequent image.¹

Given the temporal dynamics of all biological phenomena, time lapse experiments are a key part of biomedical research. Such experiments result in thousands of biomedical images displaying minute changes between frames. Automated region tracking naturally arises as the most efficient and effective means of analyzing experiments of this nature.

1.2 Existing Work

Region tracking can be approached in several ways. Three distinct classes of region tracking algorithms are those that utilise region information, those that utilise boundary information, and those that do both. Region based approaches track a region using its feature points and edge segmentation. This technique depends on assumptions about the shape and movement of the region and is most effective when the shape of the region remains relatively constant between frames and the motion is translational. A boundary based approach uses contour evolution and does not require assumptions about the shape other than smoothness. This technique considers only the boundary and not the region itself and operates under the assumption that the contrast between the region and the background is significant, which may not always be the case.

Algorithms exist that combine region and boundary-based approaches. This method first uses the region information and then computes the motion field to determine the position of the region in the subsequent frame. Because the approach is independent of topology, it can be used to track a region as it splits or merges with another region. The drawback is that the accuracy of this approach is limited when the motion is significant.

1.3 Introduction to Cell Tracking

As discussed above, many existing region tracking algorithms rely on underlying assumptions and a-priori information. These assumptions include that the shape, boundaries, intensity, position and background remain constant between images. Many existing algorithms also require motion computation. Consider the example of tracking a specific cell of interest between a sequence of microscopic images. The cell may grow, deform and move, and different cells may move in and out of the frame passing behind or in front of the cell being tracked. This is an example in which a more rigorous mathematical tracking approach is required. To illustrate

¹All sources cited throughout the report were evaluated using the C.R.A.P. test. This analysis tested the currency, reliability, authority and purpose of each source. All information presented in the sources is objective and conflicting results are discussed when necessary.

the types of images that will be tracked, three time stamps of a microscopic video of four stem cells are shown in Figures 1.1, 1.2 and 1.3.

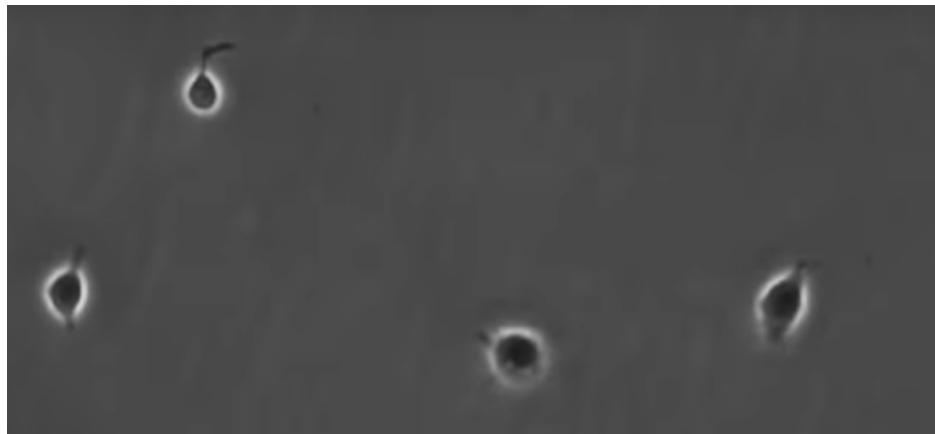


Figure 1.1: Initial image of 4 stem cells [1]

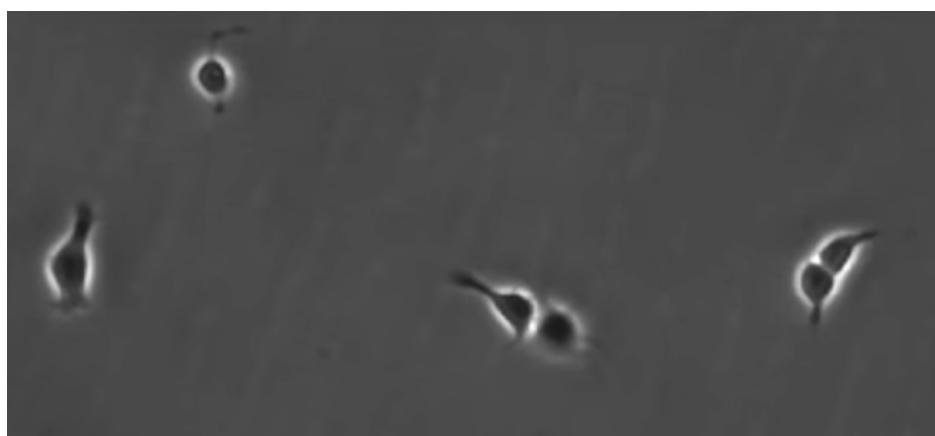


Figure 1.2: Image of the cells a few seconds later, beginning to multiply [1]

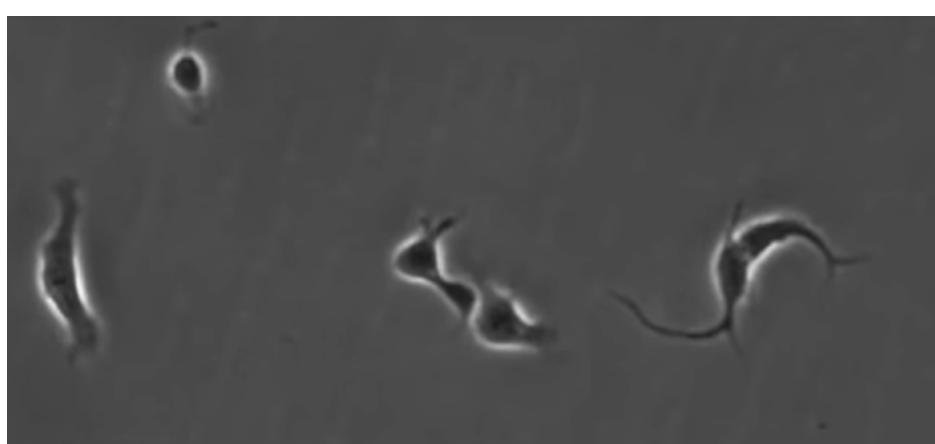


Figure 1.3: Image of the cells a few more seconds later, continuing multiplying [1]

Cell movement and proliferation are important processes to understand to gain a deeper knowl-

edge of tissue development and disease [2]. Optical microscopy is the most appropriate way to study these processes. The visualization of cells through microscopy is made possible using imaging techniques such as phase contrast and differential interference contrast microscopy, which makes cells visible without the need of external markers. Studying time-lapse videos of cells captured using optical microscopy poses the problems of accurately delineating cell boundaries and tracking the movement of chosen cells over time. Automated region tracking provides means by which a target cell can be tracked throughout cell-lineage as it divides.

The Cell Tracking Challenge – begun in 2012 – is an ongoing initiative aimed at advancing the development of cell-tracking algorithms [3]. Since 2012, 32 algorithms have been submitted, and 13 cellular imaging data sets have been made publicly available. A collaborative evaluation, conducted by 50 researchers, of all contest submissions concluded that no existing methods created fully provide the precision needed for usable tracking technology [2].

2 Problem Description

2.1 Problem Formulation

Consider two images $I_0, I_1 : \Omega \rightarrow \mathbb{R}$. Define $R_0 \subset \Omega$ as the region of interest in the image domain in image I_0 . The goal of the region tracking algorithm is to recover the region of interest, R_1 , in the subsequent image, I_1 . Define a functional (a function that takes a function as an input) $E_{(R_0, I_0, I_1)} : C \rightarrow \mathbb{R}$ where C is the set of all candidate subsets of Ω . This functional can be designed such that its unique global minimizer is R_1 . The task of “tracking” R_0 , from I_0 to I_1 , is then equivalent to the problem of finding R_1 that minimizes $E_{(R_0, I_0, I_1)}$ over C . This formulation of the region tracking problem is known as the variational approach and was first brought to computer vision by Mumford and Shah in 1989 [4].

2.2 Problem Statement

The aim of the project is to design and test a region tracking algorithm using the variational calculus approach. A region tracking functional will be designed and partial differential equations will be derived that lead to the minimizer of the functional. The functional and minimizer will be validated through computer implementation and evaluated on image sequences.

The proposed region tracking algorithm will be used to track cells over a sequence of microscopic images. This thesis will specifically focus on the tracking of stem cells in an image sequence and its potential to advance cell imaging and analysis research. Understanding stem cell mobility and how stem cells differentiate in their surroundings is crucial in understanding many biological processes.²

2.3 An Overview of Stem Cell Research

There are approximately 200 different types of cells in the human body, each serving a unique and specialized function [5]. Among these cell varieties are blood cells, bone cells, nerve cells, skin cells, muscle cells and fat cells. Stem cells are undifferentiated cells and have the potential to develop into many different types of cells. Stem cells fall into two categories: 1) embryonic stem cells and 2) adult stem cells. Embryonic cells are derived from unspecialized cells in an embryo and are pluripotent, meaning the cells can develop into any type of cell. Adult stem cells are found throughout the body and differentiate to replace dying cells and repair dying tissue. Adult stem cells are not pluripotent – the types of cells they can develop into are limited to cell types of their tissue of origin. Stem cell research has three main streams: understanding how disease occurs, generating new tissues to replace damaged tissue, and the testing of new drugs. A greater understanding of stem cell mobility, growth, and multiplication as a function of time would significantly impact all branches of stem cell research.

An example of one potential avenue of regenerative stem cell therapy is in the treatment of Alzheimer’s disease. Alzheimer’s is a neurological disorder that causes brain tissue to die. This phenomenon results in a steady decline in the patient’s thought processing, memory, and social skills. Alzheimer’s is the most common cause of dementia. The brain shrinkage caused by the decay in tissue is shown in Figure 2.1. At this time, no stem cell treatments have been approved

²Several potential applications of a region-tracking algorithm were considered ranging from vehicle applications, to anatomical/bio-mechanical applications to environmental applications in the field of global warming. All potential application areas were evaluated on their ability to result in positive social, environmental and economic impacts, how well the application was suited to this rigorous mathematical approach, and the level of difficulty.

for Alzheimer's disease. The biggest challenge is posed by the variety of neurons found in the human brain. In order to successfully treat the disease, stem cells distributed to damaged tissue must develop into the correct type of neuron and integrate with existing neurons in the environment. The mechanics by which stem cells specialize and spread is relatively unknown.



Figure 2.1: Healthy brain compared to the brain of a patient with Alzheimer's Disease [6]

3 Mathematical Background

3.1 Introduction

The following section presents the mathematical foundation used to construct the region tracking algorithm. This mathematical foundation is comprised of calculus of variations, differential geometry and curve shortening. The following theory can be referenced in articles by Grayson, Gage and Hamilton in the Journal of Differential Geometry [7], [8], [9].

The following list overviews the assumptions and methods included in this section.

1. Calculus of variations methods from Gelfand and Fomin are used to find a necessary condition for the minimizer of the energy functional E [9].
2. Using this necessary condition for a minimizer, we apply gradient descent in function space as a means of evolving a curve towards its minimizer. This results in a partial differential equation as a means of minimizing E .
3. The solution to the partial differential equation may converge to a minimum, a saddle point or diverge as $t \rightarrow \infty$. The outcome depends on the initial curve $\gamma(s, t = 0)$, where $s \mapsto \gamma(s, t)$, with $s \in [0, 1]$ is the parameterization of γ in the plane. In this report, it is assumed that given $\gamma(s, 0)$ (i.e., the boundary of region R_0), $\gamma(s, t)$ will converge to a critical point γ^* , of E , as $t \rightarrow \infty$. It is assumed that γ^* is the local minimum of E [10].
4. Gradient descent constructs a family of curves $(\gamma(s, t))_{t \geq 0}$, indexed by t , so that as $t \rightarrow \infty$, a minimizing curve γ^* is approached. Gradient descent encounters problems when the object breaks into multiple components/regions (i.e., becomes discontinuous). Hence continuously parameterizing γ by s becomes insolvable.
5. To fix the discontinuity issue, the level set method is used. γ is represented as the zero level set of some higher dimensional function $u : \mathbb{R}^2 \rightarrow \mathbb{R}$
 - (a) $u(x, y) \geq 0$ indicates that (x, y) is inside γ
 - (b) $u(x, y) < 0$ indicates that (x, y) is outside of γ
6. Given a family of curves $\gamma(s, t)$, the problem becomes how to evolve u as a function of t so that the zero level set $u : \mathbb{R}^2 \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ represents γ for all $t \in \mathbb{R}_{\geq 0}$. The curve evolution of u , with respect to t , is derived using the fact that $u(\gamma(s, t), t) = 0$ for all $s \in [0, 1]$ and $t \in \mathbb{R}_{\geq 0}$, and knowing the evolution of γ with respect to t .
7. As $t \rightarrow \infty$, the zero level set of the solution to the partial differential equation involving u will approach γ^*
8. A functional was developed using methods first introduced by Kass, Witkin, and Terzopoulos [11] as well as variations of the Mumford-Shah functional [4]

3.2 Notation

The following items address notation practices used throughout the thesis.

- Minimizer: Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a real-valued function defined on a set $\Omega \subset \mathbb{R}^n$. A point $\mathbf{x}^* \in \Omega$ is a *local minimizer* of f , over Ω , if there exists a positive ϵ such that $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for all $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$ and $\|\mathbf{x} - \mathbf{x}^*\| < \epsilon$. A point $\mathbf{x}^* \in \Omega$ is a *global minimizer* of f over Ω if $f(\mathbf{x}) \geq f(\mathbf{x}^*)$, for all $\mathbf{x} \in \Omega \setminus \{\mathbf{x}^*\}$ [12].

- Continuous Functions: The notation $f \in C^n([a, b]; \mathbb{R})$ for $n \in \mathbb{N}$ will be used throughout this report. In other words $f : [a, b] \rightarrow \mathbb{R}$ is n times continuously differentiable or f is a C^n function. If $n = 1$, then the function f is continuous.
- Smooth Functions: A function is smooth if it is C^∞ .
- Admissible Functions/Curves: We say a function $f : [a, b] \rightarrow \mathbb{R}$, $x \mapsto f(x)$ is an *admissible function* or *admissible curve* if the function is contained in the appropriate class of functions and satisfies given boundary conditions [9].
- Set of Admissible Functions/Curves: The set of all C^∞ functions from the bounded interval $[a, b]$ to the real line \mathbb{R} with end conditions $f(a) = \alpha$ and $f(b) = \beta$ is written as

$$\mathcal{C} = \{f \in C^\infty([a, b]; \mathbb{R}) | f(a) = \alpha \text{ and } f(b) = \beta\}$$

- Gradient: If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is differentiable, then the gradient of f , denoted by ∇f is defined by

$$\nabla f(\mathbf{x}) = \left(\frac{\partial}{\partial x_1} f(\mathbf{x}), \dots, \frac{\partial}{\partial x_n} f(\mathbf{x}) \right)^T$$

- Level Set: The level set of function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at level c is the set of points $\mathbf{x} \in \mathbb{R}^n$ satisfying $f(\mathbf{x}) = c$ for some c . A point \mathbf{x}_0 is on the level set, at some level c if $f(\mathbf{x}_0) = c$. The vector $\nabla f(\mathbf{x}_0)$ is orthogonal/normal to the tangent vector of an arbitrary smooth curve passing through the point \mathbf{x}_0 on the level set $f(\mathbf{x}) = c$, provided that $\nabla f(\mathbf{x}_0) \neq \mathbf{0}$. This means that the direction of the maximum rate of increase of f at a point is orthogonal to the level set of f through that point [12].
- Functional: A functional maps a function (or curve) from some function space to some number on the real line. The energy functional $E[f]$ is defined by the following

$$E[f] = \int_a^b L(x, f(x), f'(x)) dx. \quad (3.1)$$

This will be referenced throughout this report where $f(x)$ is smooth on $[a, b] \subset \mathbb{R}$, with boundary conditions $f(a) = \alpha$ and $f(b) = \beta$. Taking a closer look at (3.1), the function $L : \mathbb{R}^3 \rightarrow \mathbb{R}$, $(x, f, f') \mapsto L(x, f, f')$ is a C^∞ function and is referred to as the Lagrangian.

- Variation: Given a functional $E[f]$, the *variation* (or *differential*) of a functional $E[f]$ is defined by

$$\frac{\delta E[h]}{\delta t} = \lim_{t \rightarrow 0} \frac{E[f + th] - E[f]}{t} = \left. \frac{\partial}{\partial t} \right|_{t=0} E[f + th]$$

where h should be such a C^∞ function and should satisfy $h(a) = h(b) = 0$.

3.3 Calculus of Variations

We are interested in finding extrema³ (maximum or minimum points) of functionals of the form (3.1); this is called the calculus of variations. The calculus of variations has numerous applications in different branches of mathematics, physics and engineering. These include finding the shortest plane curve which joins two points together, the brachistochrone problem, Lagrangian mechanics, minimal surfaces, geodesics and optimal control theory. Another use for the calculus

³A word introduced by Paul du Bois-Reymond [13]

of variations, which will be the centre of this report, is image segmentation.

The next theorem will sort out the problem of determining a function (or curve) which minimizes the energy functional (3.1).

Theorem 3.1 (Necessary condition for a minimizer [9]). *Let $E : \mathcal{C} \rightarrow \mathbb{R}$ be a functional defined by*

$$E[u] = \int_a^b L(x, u(x), u_x(x)) dx \quad (3.2)$$

where $u \in C^\infty([a, b]; \mathbb{R})$ with boundary conditions $u(a) = \alpha$ and $u(b) = \beta$. If u is a local minimum of (3.2), it satisfies the Euler-Lagrange equation

$$\nabla E[u] = \frac{\partial L}{\partial u} - \frac{d}{dx} \left(\frac{\partial L}{\partial u_x} \right) = 0. \quad (3.3)$$

Proof. Note that there is an egregious abuse of notation in following proof. The partial derivatives are being evaluated at $(x, u + \epsilon v, u_x + \epsilon v_x)$ which is not being displayed to make the notation less cumbersome.

We assume that the curve $u = u(x)$ in \mathcal{C} minimizes (3.2). Since u minimizes E , then any slight perturbation of the path, $u + \epsilon v$ will not decrease E . Let $v \in C^\infty([a, b]; \mathbb{R})$ such that $v(a) = v(b) = 0$. Then $u + \epsilon v \in \mathcal{C}$ and furthermore $E[u] \leq E[u + \epsilon v]$ for all $\epsilon > 0$. Using this new function, rewrite (3.2) as

$$E[u + \epsilon v] = \int_a^b L(x, u(x) + \epsilon v(x), u_x(x) + \epsilon v_x(x)) dx. \quad (3.4)$$

Hence (3.4) is differentiable on \mathbb{R} and furthermore, 0 is a minimizer of (3.4). To check this, write

$$\begin{aligned} \frac{d}{d\epsilon} \Big|_{\epsilon=0} E[u + \epsilon v] &= \frac{d}{d\epsilon} \Big|_{\epsilon=0} \int_a^b L(x, u + \epsilon v, u_x + \epsilon v_x) dx \\ &= \int_a^b \frac{d}{d\epsilon} \Big|_{\epsilon=0} L(x, u + \epsilon v, u_x + \epsilon v_x) dx \\ &= \int_a^b \left(\frac{\partial L}{\partial u} v + \frac{\partial L}{\partial u_x} \frac{\partial v}{\partial x} \right) dx \\ &= \int_a^b \frac{\partial L}{\partial u} v dx + \int_a^b \frac{\partial L}{\partial u_x} \frac{\partial v}{\partial x} dx \\ &= \int_a^b \frac{\partial L}{\partial u} v dx + \left[\frac{\partial L}{\partial u_x} v \right]_a^b - \int_a^b \frac{d}{dx} \left(\frac{\partial L}{\partial u_x} \right) v dx \\ &= \int_a^b \frac{\partial L}{\partial u} v dx + \frac{\partial L}{\partial u_x} v(b) - \frac{\partial L}{\partial u_x} v(a) - \int_a^b \frac{d}{dx} \left(\frac{\partial L}{\partial u_x} \right) v dx \\ &= \int_a^b \frac{\partial L}{\partial u} v dx - \int_a^b \frac{d}{dx} \left(\frac{\partial L}{\partial u_x} \right) v dx \\ &= \int_a^b \left(\frac{\partial L}{\partial u} - \frac{d}{dx} \left(\frac{\partial L}{\partial u_x} \right) \right) v dx \\ &= 0 \end{aligned} \quad (3.5)$$

where (3.5) is obtained using integration by parts and (3.6) is obtained by the assumption that $v(a) = 0$ and $v(b) = 0$. ■

We reiterate that the Euler-Lagrange equation is derived from the necessary condition for a minimum (setting of a first derivative to zero). In general, one can not distinguish from this necessary condition which points are maxima, minima or any other such values. Therefore, all solutions of the Euler-Lagrange equation are referred to as extrema; doing this recognizes the fact not all extrema will be a minimum [14].

This result can be extended from curves to surfaces. Consider the surface $u = u(x, y)$ defined on $R \subset \Omega$ where u takes values on ∂R and u is a C^∞ surface.

Theorem 3.2. *Consider an energy functional E of the form*

$$E[u] = \iint_R L(x, y, u, u_x, u_y) dx dy \quad (3.7)$$

where u_x and u_y are the partial derivatives of u with respect to x and y , respectively, of the surface $u = u(x, y)$. If u is a local minimum of (3.7), it satisfies the second order Euler Lagrange equation

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} = 0. \quad (3.8)$$

Proof. Similar to the proof of Theorem 3.1, we perturb an arbitrary C^∞ surface $v = v(x, y)$ which vanishes on ∂R . With these conditions, if u is a minimizer, then $E[u + \epsilon v]$ will have a minimum at $\epsilon = 0$, and hence

$$\left. \frac{d}{d\epsilon} \right|_{\epsilon=0} E[u + \epsilon v] = 0. \quad (3.9)$$

When computing (3.9), we assume all functions are C^∞ so that the derivative can be brought inside the integral and then the chain rule can be applied. At $\epsilon = 0$, we have

$$\begin{aligned} \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} E[u + \epsilon v] &= \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} \iint_R L(x, y, u + \epsilon v, u_x + \epsilon v_x, u_y + \epsilon v_y) dx dy \\ &= \iint_R \left. \frac{d}{d\epsilon} \right|_{\epsilon=0} L(x, y, u + \epsilon v, u_x + \epsilon v_x, u_y + \epsilon v_y) dx dy \\ &= \iint_R \frac{\partial L}{\partial u} v + \frac{\partial L}{\partial u_x} \frac{\partial v}{\partial x} + \frac{\partial L}{\partial u_y} \frac{\partial v}{\partial y} dx dy \\ &= \iint_R \frac{\partial L}{\partial u} v dx dy + \iint_R \frac{\partial L}{\partial u_x} \frac{\partial v}{\partial x} dx dy + \frac{\partial L}{\partial u_y} \frac{\partial v}{\partial y} dx dy \\ &= \iint_R \frac{\partial L}{\partial u} v dx dy + \iint_R \frac{\partial}{\partial x} \left(\frac{\partial L}{\partial u_x} v \right) + \frac{\partial}{\partial y} \left(\frac{\partial L}{\partial u_y} v \right) dx dy \\ &\quad - \iint_R \left(\frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} \right) v dx dy \end{aligned} \quad (3.10)$$

$$\begin{aligned} &= \iint_R \frac{\partial L}{\partial u} v dx dy + \oint_{\partial R} \left(\frac{\partial L}{\partial u_x} v dy - \frac{\partial L}{\partial u_y} v dx \right) \\ &\quad - \iint_R \left(\frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} \right) v dx dy \end{aligned} \quad (3.11)$$

$$\begin{aligned} &= \iint_R \frac{\partial L}{\partial u} v dx dy - \iint_R \left(\frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} + \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} \right) v dx dy \\ &= \iint_R \left(\frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} \right) v dx dy \end{aligned} \quad (3.12)$$

where (3.10) is a result of integration by parts and (3.11) and is a result of Green's Theorem 3.4. The term in the parentheses in (3.12) is the first variation

$$\nabla E[u] = \frac{\partial L}{\partial u} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y}$$

which vanishes at an admissible function. Therefore the minimizer $u(x, y)$ satisfies the second order Euler Lagrange equation

$$\frac{\partial L}{\partial z} - \frac{\partial}{\partial x} \frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y} \frac{\partial L}{\partial u_y} = 0. \quad \blacksquare$$

3.4 Gradient Descent

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function and let $\mathbf{x}_0 \in \mathbb{R}^n$ be a minimizer of f , i.e., $f(\mathbf{x}_0) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$. Since \mathbf{x}_0 is a minimizer of f , then $\nabla f(\mathbf{x}_0) = 0$. Next define a C^∞ function $\mathbf{x} : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ by

$$\dot{\mathbf{x}}(t) = -\nabla f(\mathbf{x}(t)) \quad (3.13)$$

for all $t \in \mathbb{R}_{\geq 0}$. The above expression is the gradient descent ordinary differential equation, also known as gradient flow or steepest descent curve. The $\nabla f(\mathbf{x})$ points in the direction of maximum rate of increase of f at the point \mathbf{x} whereas $-\nabla f(\mathbf{x})$ points in the direction of the maximum rate of decrease of f at the point \mathbf{x} . Therefore, the direction of $-\nabla f(\mathbf{x})$ is an appropriate direction to search to find a function minimizer [12]. This concept is illustrated in Figure 3.1. The purpose of (3.13) is to evolve \mathbf{x} toward the minimum of f , by evolving \mathbf{x} against the gradient, since f dissipates along the flow of (3.13). This is equivalent to saying the following: as t increases, the solution to (3.13) will approach the minimizer \mathbf{x}_0 of f , provided that f is bounded from below. Letting \mathbf{x} be a solution to (3.13) we see that for all $t \in \mathbb{R}_{\geq 0}$, we have the following.

$$\begin{aligned} \frac{d}{dt}(f \circ \mathbf{x})(t) &= \frac{d}{dt} f(\mathbf{x}(t)) \\ &= \langle \nabla f(\mathbf{x}(t)), \dot{\mathbf{x}}(t) \rangle \\ &= \langle \nabla f(\mathbf{x}(t)), -\nabla f(\mathbf{x}(t)) \rangle \\ &= -\langle \nabla f(\mathbf{x}(t)), \nabla f(\mathbf{x}(t)) \rangle \\ &= -\|\nabla f(\mathbf{x}(t))\|^2 \\ &\leq 0 \end{aligned}$$

Where $\|\cdot\|$ denotes the Euclidean norm on \mathbb{R}^n . This shows that $f(\mathbf{x}(t))$ is a monotonically decreasing function of t .

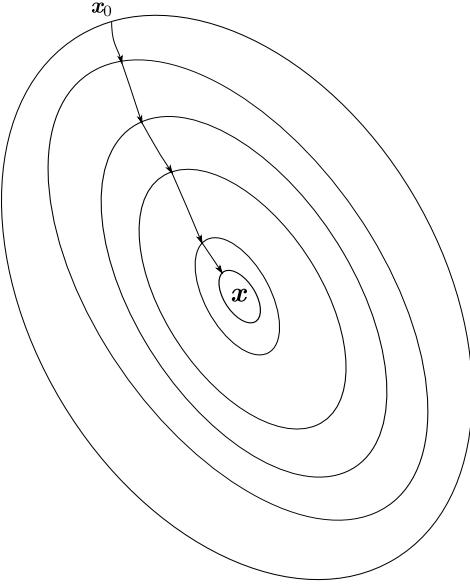


Figure 3.1: Visualization of gradient descent

Let $\mathbf{x}^{(0)}$ be a starting point and consider the point $\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$. Using Taylor's formula, we can write

$$f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) = f(\mathbf{x}^{(0)}) - \alpha \|\nabla f(\mathbf{x}^{(0)})\|^2 + o(\alpha).$$

If $\nabla f(\mathbf{x}^{(0)}) \neq \mathbf{0}$, then for some small $\alpha > 0$,

$$f(\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})) < f(\mathbf{x}^{(0)}).$$

This means that if we are searching for a minimizer, the point $\mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$ is better than the point $\mathbf{x}^{(0)}$. Now suppose we start at a point $\mathbf{x}^{(k)}$ and move by $-\alpha_k \nabla f(\mathbf{x}^{(k)})$ to find the next point $\mathbf{x}^{(k+1)}$ for $\alpha_k > 0$ where α_k is called the step size. This procedure gives us the gradient descent algorithm

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)}).$$

The gradient changes as the search for the minimizer proceeds and goes to zero as the minimizer is approached [12].

We now move on to discuss gradient descent in the space of functions.

Proposition 3.3. *Let $L : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a C^∞ function. Define $f \in C^\infty([a, b] \times \mathbb{R}_{\geq 0}; \mathbb{R})$, $(x, t) \mapsto f(x, t)$ such that $f(a, t) = \alpha$ and $f(b, t) = \beta$ for all $t \in \mathbb{R}_{\geq 0}$. If f satisfies the Euler-Lagrange gradient descent equation*

$$\frac{\partial}{\partial t} f(x, t) = - \left(\frac{\partial L}{\partial f} \Big|_{(x, f(x, t), f'(x, t))} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \Big|_{(x, f(x, t), f'(x, t))} \right) \right), \quad (3.14)$$

then for all $t \in \mathbb{R}_{\geq 0}$ and $x \in [a, b]$,

$$\frac{d}{dt} E[f(x, t)] \leq 0.$$

Proof. Since f is a solution to (3.14), we can write

$$\frac{d}{dt} E[f(x, t)] = \frac{d}{dt} \int_a^b L(x, f(x, t), f'(x, t)) dx$$

$$\begin{aligned}
&= \int_a^b \frac{d}{dt} L(x, f(x, t), f'(x, t)) dx \\
&= \int_a^b \left(\frac{\partial L}{\partial f} \frac{\partial f}{\partial t} + \frac{\partial L}{\partial f'} \frac{\partial f'}{\partial t} \right) dx \\
&= \int_a^b \left(\frac{\partial L}{\partial f} \frac{\partial f}{\partial t} + \frac{\partial L}{\partial f'} \frac{d}{dx} \left(\frac{\partial f}{\partial t} \right) \right) dx \\
&= \int_a^b \left(\frac{\partial L}{\partial f} \frac{\partial f}{\partial t} \right) dx + \int_a^b \left(\frac{\partial L}{\partial f'} \frac{d}{dx} \left(\frac{\partial f}{\partial t} \right) \right) dx \\
&= \int_a^b \left(\frac{\partial L}{\partial f} \frac{\partial f}{\partial t} \right) dx + \left[\frac{\partial L}{\partial z} \frac{\partial f}{\partial t} \right]_a^b - \int_a^b \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \frac{\partial f}{\partial t} dx \\
&= \int_a^b \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \right) \left(- \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \right) \right) dx \\
&= - \int_a^b \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \right) \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \right) dx \\
&= - \int_a^b \left(\frac{\partial L}{\partial f} - \frac{d}{dx} \left(\frac{\partial L}{\partial f'} \right) \right)^2 dx \\
&\leq 0
\end{aligned}$$

for all $t \in \mathbb{R}_{\geq 0}$. Therefore $t \mapsto E[f(x, t)]$ is monotonically decreasing. \blacksquare

The following sections introduce applications of Theorems 3.1 and 3.2, as well as Proposition 3.3.

3.4.1 Minimizing Contour Integrals

Let $\gamma : [0, 1] \times [0, T] \rightarrow \mathbb{R}^2$ be a C^∞ curve parameterized by $(s, t) \mapsto \gamma(s, t) = (x(s, t), y(s, t))$ with $\gamma(0, t = 0) = \gamma(1, t = T)$, i.e. the initial and terminal points coincide. In other words, let $\gamma = \gamma(s, t)$ be a simple, smooth, closed curve in \mathbb{R}^2 . The notation $\gamma(s, t)$ is simply a point on the curve at some position and time where s is the arc-length parameter and t is the time. We will redefine the set \mathcal{C} as

$$\mathcal{C} = \{\gamma \in C^\infty([0, 1] \times \mathbb{R}_{\geq 0}; \mathbb{R}^2) \mid \gamma(0, t = 0) = \gamma(1, t = T)\}.$$

Consider the functional $E : \mathcal{C} \rightarrow \mathbb{R}$ given by

$$E[\gamma] = \oint_{\gamma} ds \tag{3.15}$$

where \circlearrowleft in the integral indicates that the integral is to be taken in the counter clockwise direction (positive orientation). Let $(\gamma(\cdot, t))_{t \geq 0}$ be the one-parameter family of curves generated by moving $\gamma = \gamma(s, t)$ along the curve. To determine a curve $\gamma^* = \gamma^*(s, t)$ which minimizes (3.15), embed γ in the one-parameter family so that $\gamma \rightarrow \gamma^*$ as $t \rightarrow \infty$. This one-parameter family is achieved using (3.14) from Proposition 3.3. We can rewrite (3.15) as

$$E[\gamma(s, t)] = \int_0^1 \|\dot{\gamma}(s, t)\| ds = \int_0^1 (\dot{x}^2(s, t) + \dot{y}^2(s, t))^{1/2} ds$$

where the Lagrangian is $L(s, x, y, \dot{x}, \dot{y}) = \sqrt{\dot{x}^2 + \dot{y}^2}$. The “ $\dot{\gamma}$ ” represents differentiation with respect to the s variable. Note that $\dot{\gamma}(s, t)$ is the tangent to $\gamma(s, t)$ (γ_s is the change in the

position as the arc length changes), $\ddot{\gamma}(s, t)$ is the change in the tangent as the arc-length changes and γ_t is the motion of a curve over time.

When writing the gradient descent equations for $\gamma(s, t)$ defined in Proposition 3.3, we get a system of partial differential equations. The gradient descent partial differential equation for the $x = x(s, t)$ component of γ is as follows.

$$\begin{aligned}\frac{\partial x}{\partial t} &= -\left(\frac{\partial L}{\partial x} - \frac{d}{ds}\left(\frac{\partial L}{\partial \dot{x}}\right)\right) \\ &= \frac{d}{ds}\left(\frac{\partial L}{\partial \dot{x}}\right) \\ &= \frac{d}{ds}\left(\frac{\dot{x}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}\right) \\ &= \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}(-\dot{y})\end{aligned}\tag{3.16}$$

The gradient descent partial differential equation for the $y = y(s, t)$ component of γ is as follows.

$$\begin{aligned}\frac{\partial y}{\partial t} &= -\left(\frac{\partial L}{\partial y} - \frac{d}{ds}\left(\frac{\partial L}{\partial \dot{y}}\right)\right) \\ &= \frac{d}{ds}\left(\frac{\partial L}{\partial \dot{y}}\right) \\ &= \frac{d}{ds}\left(\frac{\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{1/2}}\right) \\ &= \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}\dot{x}\end{aligned}\tag{3.17}$$

Combining (3.16) and (3.17) yields

$$\begin{aligned}\frac{\partial \gamma}{\partial t} &= \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}\right) \\ &= \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}(-\dot{y}, \dot{x}) \\ &= \kappa(-\dot{y}, \dot{x})\end{aligned}$$

where $\kappa = \kappa(s, t)$ is the parameterization of the curvature of the path γ . The curvature is defined as

$$\kappa = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}.$$

The position of the curve is $\gamma = (x, y)$ and the velocity of the curve is $\dot{\gamma} = (\dot{x}, \dot{y})$. Define the normal vector $\mathbf{n} = \mathbf{n}(s, t)$ to the curve γ as $\mathbf{n} = (-\dot{y}, \dot{x})$. By taking the Euclidean inner product on \mathbb{R}^2 between \mathbf{n} and $\dot{\gamma}$ one sees that $\langle \dot{\gamma}, \mathbf{n} \rangle = 0$. Therefore, the normal vector is orthogonal to the velocity vector. The orientation of the normal is determined by extending the vectors to \mathbb{R}^3 and taking their cross product

$$\begin{aligned}(\dot{x}, \dot{y}, 0) \times (-\dot{y}, \dot{x}, 0) &= \det \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \\ \dot{x} & \dot{y} & 0 \\ -\dot{y} & \dot{x} & 0 \end{bmatrix} \\ &= \det \begin{bmatrix} \dot{y} & 0 \\ \dot{x} & 0 \end{bmatrix} \hat{i} - \det \begin{bmatrix} \dot{x} & 0 \\ -\dot{y} & 0 \end{bmatrix} \hat{j} + \det \begin{bmatrix} \dot{x} & \dot{y} \\ -\dot{y} & \dot{x} \end{bmatrix} \hat{k}\end{aligned}$$

$$\begin{aligned}
&= 0\hat{\mathbf{i}} + 0\hat{\mathbf{j}} + (\dot{x}^2 + \dot{y}^2)\hat{\mathbf{k}} \\
&= (0, 0, \dot{x}^2 + \dot{y}^2).
\end{aligned}$$

We notice that the cross product is pointing in the positive z direction. By the right-hand rule we define \mathbf{n} as the inward pointing normal vector.

The Euler-Lagrange gradient decent equation can then be written as

$$\frac{\partial \gamma}{\partial t} = \frac{\partial^2 \gamma}{\partial s^2} = \kappa \mathbf{n}. \quad (3.18)$$

The above equation is known as the mean curvature flow. An interpretation of (3.18) is as follows. As t increases, the curve will evolve so that the regions of positive curvature will be pushed outward, while regions of negative curvature will be pushed inward. This process is illustrated in Figure 3.2. The mean curvature flow is also known as the geometric heat equation which describes a curve travelling along its normal direction with a velocity equal to its curvature. A result from the study of curve evolution states that the geometric shape of the curve is only affected by the normal component of the curve's velocity. The tangential component does not effect the geometric shape of the propagating curve; only the parameterization [15].

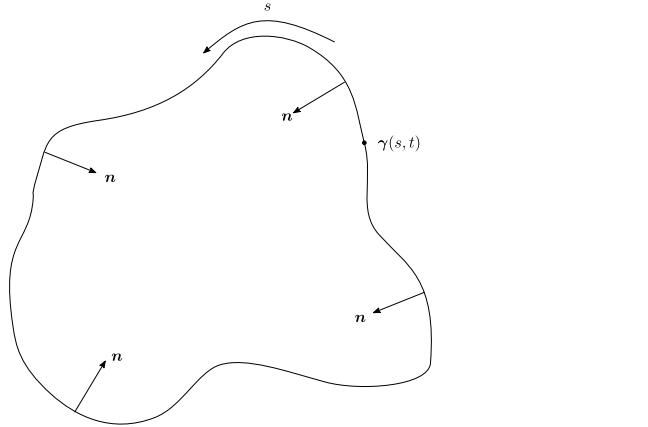


Figure 3.2: Propagating curve

We will now show how to use the Euler-Lagrange equations to minimize surface integrals.

3.4.2 Minimizing Surface Integrals

When minimizing surface integrals we consider the energy functional. Let $F : \mathbb{R}^2 \rightarrow \mathbb{R}$ and consider the energy functional defined by

$$E[\gamma] = \iint_{R_\gamma} F(x, y) dx dy \quad (3.19)$$

where R_γ is a subset of \mathbb{R}^2 enclosed by γ . This result is due to Theorem A.2.

Using Green's Theorem, surface integrals can be converted into contour integrals, making it possible to minimize the integrals using the Euler-Lagrange equations.

Theorem 3.4 (Green's Theorem). *Given functions $P : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $Q : \mathbb{R}^2 \rightarrow \mathbb{R}$ defined on an open region containing R_γ , and with continuous partial derivatives, then*

$$\iint_{R_\gamma} \left(\frac{\partial}{\partial x} Q(x, y) - \frac{\partial}{\partial y} P(x, y) \right) dx dy = \oint_\gamma P(x, y) dx + Q(x, y) dy.$$

Let $P : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $Q : \mathbb{R}^2 \rightarrow \mathbb{R}$ be defined as

$$\begin{aligned} P(x, y) &= -\frac{1}{2} \int_{y_0}^y F(x, z) dz \\ Q(x, y) &= \frac{1}{2} \int_{x_0}^x F(z, y) dz \end{aligned}$$

for $(x_0, y_0) \in \mathbb{R}^2$. Our choice of P and Q allows us to write

$$\begin{aligned} \frac{\partial}{\partial y} P(x, y) &= -\frac{1}{2} F(x, y) \\ \frac{\partial}{\partial x} Q(x, y) &= \frac{1}{2} F(x, y). \end{aligned}$$

This means that F can be written as

$$F(x, y) = \frac{\partial}{\partial x} Q(x, y) - \frac{\partial}{\partial y} P(x, y).$$

By Green's Theorem 3.4, we can write (3.19) as

$$\begin{aligned} E[\gamma] &= \iint_{R_\gamma} F(x, y) dx dy \\ &= \iint_{R_\gamma} \left(\frac{\partial}{\partial x} Q(x, y) - \frac{\partial}{\partial y} P(x, y) \right) dx dy \\ &= \oint_\gamma P(x, y) dx + Q(x, y) dy \\ &= \int_0^1 (P(x, y) \dot{x} + Q(x, y) \dot{y}) ds. \end{aligned}$$

It is now possible to minimize the contour integral using the Euler-Lagrange equations, noting that the Lagrangian is $L = P(x, y) \dot{x} + Q(x, y) \dot{y}$. The Euler-Lagrange gradient descent equation for x is

$$\begin{aligned} \frac{\partial x}{\partial t} &= - \left(\frac{\partial L}{\partial x} - \frac{d}{ds} \left(\frac{\partial L}{\partial \dot{x}} \right) \right) \\ &= - \left(\frac{\partial}{\partial x} P(x, y) \dot{x} + \frac{\partial}{\partial x} Q(x, y) \dot{y} - \frac{d}{ds} P(x, y) \right) \\ &= - \left(\frac{\partial}{\partial x} P(x, y) \dot{x} + \frac{\partial}{\partial x} Q(x, y) \dot{y} - \frac{\partial}{\partial x} P(x, y) \dot{x} - \frac{\partial}{\partial y} P(x, y) \dot{y} \right) \\ &= - \left(\frac{\partial}{\partial x} Q(x, y) - \frac{\partial}{\partial y} P(x, y) \right) \dot{y} \\ &= -F(x, y) \dot{y}. \end{aligned} \tag{3.20}$$

Similarly, the Euler-Lagrange gradient descent equation for y is

$$\frac{\partial y}{\partial t} = - \left(\frac{\partial L}{\partial y} - \frac{d}{ds} \left(\frac{\partial L}{\partial \dot{y}} \right) \right)$$

$$\begin{aligned}
&= - \left(\frac{\partial}{\partial y} P(x, y) \dot{y} + \frac{\partial}{\partial x} Q(x, y) \dot{y} - \frac{d}{ds} Q(x, y) \right) \\
&= - \left(\frac{\partial}{\partial x} P(x, y) \dot{x} + \frac{\partial}{\partial x} Q(x, y) \dot{y} - \frac{\partial}{\partial x} Q(x, y) \dot{x} - \frac{\partial}{\partial y} Q(x, y) \dot{y} \right) \\
&= - \left(\frac{\partial}{\partial x} P(x, y) - \frac{\partial}{\partial y} Q(x, y) \right) \dot{x} \\
&= F(x, y) \dot{x}.
\end{aligned} \tag{3.21}$$

Combining (3.20) and (3.21) produces

$$\frac{\partial \gamma}{\partial t} = F \mathbf{n} \tag{3.22}$$

which is known as the gradient descent curve evolution equation [16]. In this discussion, the speed function, F only depends on the local curvature of the curve, that is $F = F(\kappa)$. Curvature points in the direction normal to the curve γ [17].

3.5 Level Set Method

In the previous section we expressed the Euler-Lagrange descent equation in terms of the motion of a curve γ . To do this computation, we could represent γ either explicitly or implicitly. An explicit representation would involve a numeric discretization of the interval on which γ is defined, yielding a representation of γ in terms of a finite number of points. This approach can lead to significant approximation errors. Alternatively, we can represent γ implicitly by the zero level set of a function $u : \mathbb{R}^2 \rightarrow \mathbb{R}$. This approach is favourable and does not lead to approximation errors as each level set function is defined over a fixed, uniform discrete grid.

Take $(u(\gamma, t))_{t \geq 0}$ to be a family of surfaces parameterized by t . Since γ is the zero level set of u , the relation given by

$$u(\gamma, t) = 0 \tag{3.23}$$

for all $s \in [0, 1]$ and $t \in \mathbb{R}_{\geq 0}$ must be preserved. In order to preserve the relation (3.23), the equations

$$\frac{d}{ds} u(\gamma, t) = 0 \tag{3.24a}$$

$$\frac{d}{dt} u(\gamma, t) = 0 \tag{3.24b}$$

need to hold for all $s \in [0, 1]$ and all $t \in \mathbb{R}_{\geq 0}$. Throughout this report, we will use the convention that $u \geq 0$ in the interior of the curve γ and $u < 0$ on the exterior of the curve γ . Taking the derivative with respect to s of (3.23) yields

$$\begin{aligned}
\frac{d}{ds} u(\gamma, t) &= \frac{\partial}{\partial s} u(x(s, t), y(s, t), t) \\
&= \frac{\partial}{\partial x} u(\gamma, t) \frac{\partial}{\partial s} x(s, t) + \frac{\partial}{\partial y} u(\gamma, t) \frac{\partial}{\partial s} y(s, t) \\
&= \nabla u(\gamma, t) \cdot \dot{\gamma}.
\end{aligned}$$

By (3.24a), we find $\nabla u(\gamma, t) \cdot \dot{\gamma}(s, t) = 0$ implying that $u(\gamma, t)$ is orthogonal to $\dot{\gamma}(s, t)$. Similarly, taking the derivative (3.23) with respect to t gives

$$\frac{d}{dt} u(\gamma, t) = \frac{\partial}{\partial t} u(x(s, t), y(s, t), t)$$

$$\begin{aligned}
&= \frac{\partial}{\partial x} u(\gamma, t) \frac{\partial}{\partial t} x(s, t) + \frac{\partial}{\partial y} u(\gamma, t) \frac{\partial}{\partial t} y(s, t) + \frac{\partial \gamma}{\partial t} \\
&= \nabla u(\gamma, t) \frac{\partial \gamma}{\partial t} + \frac{\partial}{\partial t} u(\gamma, t).
\end{aligned}$$

By (3.24b), we can write

$$\frac{\partial}{\partial t} u(\gamma, t) = -\nabla u(\gamma, t) \frac{\partial \gamma}{\partial t}. \quad (3.25)$$

Substituting (3.22) into (3.25) yields

$$\frac{\partial}{\partial t} u(\gamma, t) = -F(\nabla u(\gamma, t) \cdot \mathbf{n}(s, t)). \quad (3.26)$$

Since γ is a level set of u , the gradient of u is orthogonal to γ . As we have used the convention that $u \geq 0$ in the interior of the curve γ and $u < 0$ on the exterior of the curve γ , the gradient of u is simply a non-negative scalar multiple of \mathbf{n} . As a result, we can rewrite (3.26) as

$$\frac{\partial}{\partial t} u(\gamma, t) = -F \|\nabla u(\gamma, t)\| \quad (3.27)$$

where (3.27) describes the curve evolution of γ .

3.5.1 Curvature in the Level Set Method

Recall the mean curvature flow equation derived in 3.4.1

$$\frac{\partial \gamma}{\partial t} = \kappa \mathbf{n}.$$

In the previous section, the function F is assumed to be independent of $\gamma(s, t) = (x(s, t), y(s, t))$. However, recalling the definition of κ ,

$$\kappa(s, t) = \frac{\dot{x}(s, t)\ddot{y}(s, t) - \ddot{x}(s, t)\dot{y}(s, t)}{(\dot{x}^2(s, t) + \dot{y}^2(s, t))^{3/2}}.$$

this is not the case. Thus, in order to use the level set method, it is necessary to express κ in terms of the level set function, $u = u(\gamma(s, t), t)$. Since ∇u is orthogonal to $\dot{\gamma} = \dot{\gamma}(s, t)$, as we showed in 3.4.1,

$$\nabla u \cdot \dot{\gamma} = \frac{\partial u}{\partial x} \dot{x} + \frac{\partial u}{\partial y} \dot{y} = 0. \quad (3.28)$$

By differentiating (3.28) with respect to s , we can then write

$$\left(\frac{\partial^2 u}{\partial x^2} \dot{x} + \frac{\partial^2 u}{\partial x \partial y} \dot{y} \right) \ddot{x} + \frac{\partial u}{\partial x} \ddot{x} + \left(\frac{\partial^2 u}{\partial x \partial y} \dot{x} + \frac{\partial^2 u}{\partial y^2} \dot{y} \right) \ddot{y} + \frac{\partial^2 u}{\partial y^2} \ddot{y} = 0. \quad (3.29)$$

By rearranging (3.29) we get

$$\frac{\partial u}{\partial x} \ddot{x} + \frac{\partial^2 u}{\partial y^2} \ddot{y} = - \left(\frac{\partial^2 u}{\partial x^2} \dot{x} + \frac{\partial^2 u}{\partial x \partial y} \dot{y} \right) \dot{x} - \left(\frac{\partial^2 u}{\partial x \partial y} \dot{x} + \frac{\partial^2 u}{\partial y^2} \dot{y} \right) \dot{y} \quad (3.30)$$

and simplifying (3.30) yields

$$\nabla u \cdot \ddot{\gamma} = -\dot{\gamma}^\top \text{Hess}(u) \dot{\gamma} \quad (3.31)$$

where $\text{Hess}(u)$ is the Hessian of u given by

$$\text{Hess}(u) = \begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial x \partial y} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix}.$$

Suppose γ is parameterized at unit speed, that is, $\|\dot{\gamma}\| = 1$ for all $t \in \mathbb{R}_{\geq 0}$. Using (3.18) and (3.27) respectively, we can rewrite the left hand side of (3.31) to get

$$\begin{aligned} \nabla u \cdot \ddot{\gamma} &= \kappa(\nabla u \cdot \mathbf{n}) \\ &= \kappa \|\nabla u\| \end{aligned}$$

and our new expression for (3.31) is

$$\kappa \|\nabla u\| = -\dot{\gamma}^\top \text{Hess}(u) \dot{\gamma}. \quad (3.32)$$

Define $\nabla u^\perp = (-u_y, u_x)$ so that ∇u^\perp is orthogonal to ∇u , and hence, ∇u is a scalar multiple of $\dot{\gamma}$. Since γ is parameterized with unit speed, we get the normalized expression for $\dot{\gamma}$

$$\dot{\gamma} = \frac{\pm \nabla u^\perp}{\|\nabla u\|} \quad (3.33)$$

where directions of γ and ∇u^\perp do not matter in the subsequent calculations. Substituting (3.33) into (3.32) and solving for κ gives us

$$\begin{aligned} \kappa \|\nabla u\| &= -\dot{\gamma}^\top \text{Hess}(u) \dot{\gamma} \\ &= -\frac{(\nabla u^\perp)^\top \text{Hess}(u) \nabla u^\perp}{\|\nabla u\|^2} \\ \implies \kappa &= -\frac{(\nabla u^\perp)^\top \text{Hess}(u) \nabla u^\perp}{\|\nabla u\|^3}. \end{aligned} \quad (3.34)$$

By multiplying out (3.34), one finds that

$$\kappa(x, y, t) = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}. \quad (3.35)$$

3.6 Discretization

An image is represented as a set of pixels uniformly distributed; as a result, we can think of an image as a grid/mesh as depicted in Figure 3.3. For simplicity, assume that Δx and Δy are uniformly spaced throughout the grid; that is, $\Delta x = \Delta y = h$ where Δx and Δy are the distances between adjacent points and where Δt is the time step. The goal of this section is to discretize (3.26). First note that (3.26) can be written as

$$\frac{\partial}{\partial t} u(\gamma, t) = \frac{\partial^2}{\partial s^2} u(\gamma, t) = -F \|\nabla u(\gamma, t)\|. \quad (3.36)$$

Discretization of (3.36) requires obtaining a finite difference scheme by invoking Taylor's theorem. There are three finite differences schemes that are commonly considered for numerical approximation: forward, backward and central difference schemes. The forward difference scheme in time is given by

$$u(x, y, t + \Delta t) = u(x, y, t) + \Delta t \frac{\partial}{\partial t} u(x, y, t) + \frac{\Delta t^2}{2!} \frac{\partial^2}{\partial t^2} u(x, y, t) + \dots$$

$$= u(x, y, t) + \Delta t \frac{\partial}{\partial t} u(x, y, t) + O(\Delta t^2) \quad (3.37)$$

where $O(\Delta t^2)$ is the error term of order Δt^2 . Rearrange (3.37) and divide by Δt to obtain

$$\begin{aligned} \Delta t \frac{\partial}{\partial t} u(x, y, t) + O(\Delta t^2) &= u(x, y, t + \Delta t) - u(x, y, t) \\ \frac{\partial}{\partial t} u(x, y, t) + O(\Delta t) &= \frac{u(x, y, t + \Delta t) - u(x, y, t)}{\Delta t} \\ \Rightarrow \frac{\partial}{\partial t} u(x, y, t) &\approx \frac{u(x, y, t + \Delta t) - u(x, y, t)}{\Delta t}. \end{aligned} \quad (3.38)$$

To complete the discretization, we evaluate (3.38) at $x_i = ih$, $y_j = jh$ and $t = n\Delta t$ to get

$$\frac{\partial}{\partial t} u(x, y, t) = \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$$

where $u_{i,j}^n$ denotes the values of the discrete solution to (3.36) at location (ih, jh) at time $n\Delta t$, see Figure 3.3. Note that the points (ih, jh) on the grid are known as either pixels, nodes or vertices. The methods to compute the remaining forward, backward and central difference equations for the various derivatives were omitted for brevity; however, we summarize the finite difference equations in Tables 3.1 and 3.2.

Table 3.1: Forward and backward difference approximations

| | Forward difference | Backward difference |
|----------------|---|---|
| $u_t(x, y, t)$ | $\frac{u(x, y, t + \Delta t) - u(x, y, t)}{\Delta t}$ | |
| $u_x(x, y, t)$ | $\frac{u(x + h, y, t) - u(x, y, t)}{h}$ | $\frac{u(x, y, t) - u(x - h, y, t)}{h}$ |
| $u_y(x, y, t)$ | $\frac{u(x, y + h, t) - u(x, y, t)}{h}$ | $\frac{u(x, y, t) - u(x, y - h, t)}{h}$ |

Table 3.2: Central difference approximations

| | Central difference |
|-------------------|--|
| $u_x(x, y, t)$ | $\frac{u(x + h, y, t) - u(x - h, y, t)}{2h}$ |
| $u_y(x, y, t)$ | $\frac{u(x, y + h, t) - u(x, y - h, t)}{2h}$ |
| $u_{xx}(x, y, t)$ | $\frac{u(x + h, y, t) - 2u(x, y, t) + u(x - h, y, t)}{h^2}$ |
| $u_{yy}(x, y, t)$ | $\frac{u(x, y + h, t) - 2u(x, y, t) + u(x, y - h, t)}{h^2}$ |
| $u_{xy}(x, y, t)$ | $\frac{u(x + h, y + h, t) - u(x + h, y - h, t) - u(x - h, y + h, t) + u(x - h, y - h, t)}{4h^2}$ |

The discretized finite difference equations are shown in Table 3.3 and Table 3.4.

Table 3.3: Discretization of the forward and backward difference approximations

| | Forward difference | Backward difference |
|----------------|--|-------------------------------------|
| $u_t(x, y, t)$ | $\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}$ | |
| $u_x(x, y, t)$ | $\frac{u_{i+1,j}^n - u_{i,j}^n}{h}$ | $\frac{u_{i,j}^n - u_{i-1,j}^n}{h}$ |
| $u_y(x, y, t)$ | $\frac{u_{i,j+1}^n - u_{i,j}^n}{h}$ | $\frac{u_{i,j}^n - u_{i,j-1}^n}{h}$ |

Table 3.4: Discretization of the central difference approximations

| | Central difference |
|-------------------|--|
| $u_x(x, y, t)$ | $\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2h}$ |
| $u_y(x, y, t)$ | $\frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h}$ |
| $u_{xx}(x, y, t)$ | $\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h^2}$ |
| $u_{yy}(x, y, t)$ | $\frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h^2}$ |
| $u_{xy}(x, y, t)$ | $\frac{u_{i+1,j+1}^n - u_{i+1,j-1}^n - u_{i-1,j+1}^n + u_{i-1,j-1}^n}{4h^2}$ |

For a robust overview of solving partial differential equations using finite differences, see [18] and [17].

Recall,

$$\kappa(x, y, t) = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}$$

The central difference approximation will be used to obtain numerical solutions to (3.35). The discrete representations of each of the partial derivatives are written below.

Substitute the forward, backward and central difference approximations from Tables 3.3 and 3.4 into (3.36) to get the iterative update scheme

$$\frac{\partial}{\partial t}u(\gamma, t) = \frac{\partial^2}{\partial s^2}u(\gamma, t)$$

$$\begin{aligned}
\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} &= \frac{u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n}{h^2} + \frac{u_{i,j-1}^n - 2u_{i,j}^n + u_{i,j+1}^n}{h^2} \\
&= \frac{u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n + u_{i+1,j}^n + u_{i,j+1}^n}{h^2} \\
u_{i,j}^{n+1} - u_{i,j}^n &= \Delta t \left(\frac{u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n + u_{i+1,j}^n + u_{i,j+1}^n}{h^2} \right) \\
&= u_{i,j}^n + \Delta t \left(\frac{u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n + u_{i+1,j}^n + u_{i,j+1}^n}{h^2} \right)
\end{aligned}$$

where $u(x, y, 0) = u_0(x, y)$ is the initial data [15], [17].

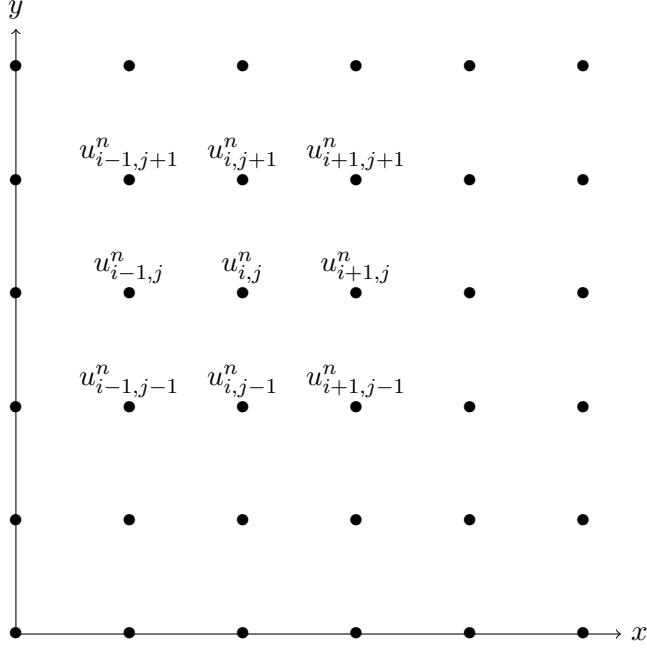


Figure 3.3: Computational grid

The method of Sethian is used to represent the gradient of u discretely. Given a functional F defined at a point (i, j) by $F_{i,j}$, the gradient of u at that point is given by

$$\|\nabla u\| = \begin{cases} \nabla^+, & \text{if } F_{i,j} > 0 \\ \nabla^-, & \text{if } F_{i,j} \leq 0 \end{cases}$$

where ∇^+ represents the forwards difference approximation and ∇^- represents the backwards difference approximation, defined in Table 3.3.

An important partial differential equation in image processing is (3.18) which is of similar flavour as the 2D heat equation. The heat equation describes the distribution of heat in a given region over time.

3.7 Validating Design Approach

Before proceeding to the next phase of the project, the design process was validated using the length functional. This entailed creating a functional that exhibited the following behaviour:

- Smooth convex curves that undergo curve-shortening flow contract to a point, as proved by Gage in 1984 [19],
- A convex curve in \mathbb{R}^2 shrinks to a point under the heat equation, as proved by Gage and Hamilton in 1986 [7], and
- Non-convex, simple curves that undergo curve-shortening flow eventually become convex, as proved by Grayson in 1987 [8].

To verify that the approach taken is correct, it must be seen that as a non-convex shape undergoes curve-shortening, the flow shall become convex and eventually contract to a point. To evaluate this, a non-convex curve was drawn and allowed to evolve under the heat equation through a computer programming simulation. The results of the implementation of mean curvature flow are shown in Figure 3.4.

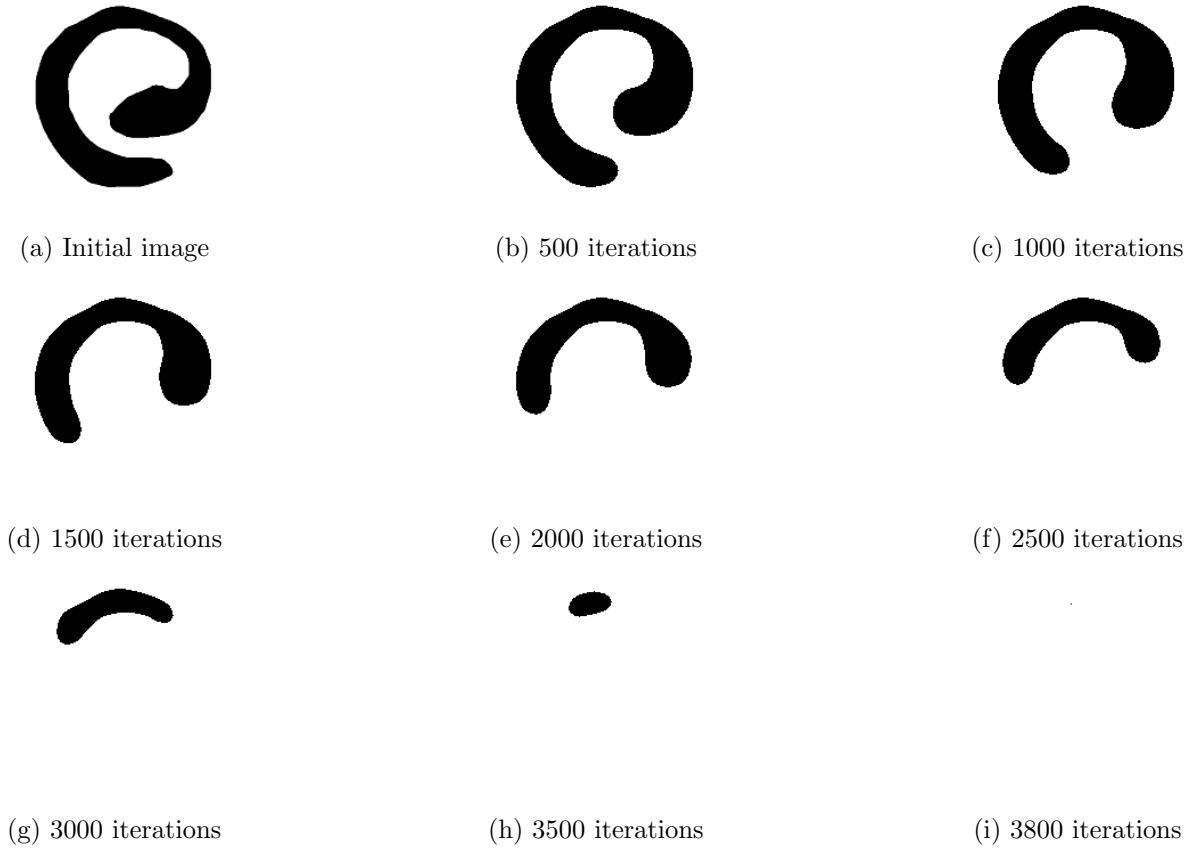


Figure 3.4: Mean curvature flow validation

As expected, the non-convex spiral becomes convex before eventually collapsing to a point. This validates the design approach taken.

4 Design Setup

4.1 Choice of Programming Language

Before proceeding with the implementation of a computationally intensive program, the coding environment was considered to ensure it was suitable for the algorithm's behaviour. For instance, each iteration of the proposed algorithm requires extensive array manipulations. As a result, the programming language must have minimal computational overhead. In addition, the algorithm requires frequent image input and output, thus, the programming environment must support this behaviour.

Two programming environments were considered: programming in a combination of C and MATLAB, and programming in Python. C language has excellent run-time speed and minimal computational overhead while MATLAB has excellent image I/O capabilities. The former would involve developing a MATLAB program which performed image I/O via MATLAB functions, while running the region tracking algorithm via embedded C code. This would require maintaining and updating code in two locations. The latter option was considered for user friendliness and functionality. Python is an easy to learn language that has a variety of open-source libraries at the users disposal. Libraries, such as CV2 and Pillow, allow the user to import and write to images. It is therefore possible to write the entire program in Python, rather than combining languages and splitting code location.

For simplicity of code maintenance and code centrality, as well as the benefit of available open-source libraries for image processing and computer vision, Python was chosen for the implementation of the region tracking algorithm. Windows OS was used for development and implementation of the algorithm.

4.2 Pseudocode

The first step of implementing the region tracking algorithm is to discretize the partial derivatives used to find a global minimizer of the designed region tracking functional. The method of discretization is introduced and discussed above in Section 3.6. The implementation of the discretization of the region tracking algorithm in a coding environment has the general form described in the following pseudocode.

The matrix A^n was used to define the region of interest, where each pixel is given one of two values: 1 if it is inside the region of interest, or -1 if it is outside the region of interest. The algorithm updates this matrix each iteration, using the defined functional, F^n , the central difference approximation to the curvature expression, $K_{i,j}^n$ and the difference operators $D_{i,j}^{0x}$ and $D_{i,j}^{0y}$ [15], [17]. The difference operators use information at $i + 1$ and $i - 1$ to compute the new value at i . The part of the term including F^n is designed to penalise deviation from time-invariant properties of the original region. The $K_{i,j}^n$ is designed to penalise the length of the curve. The design components of the algorithm are the time step, the stopping condition, the determination of the linear combination of functionals, the functional weighting within the linear combination, and the weighting of the noise term, λ_{length} .

Algorithm 1: Level Set Evolution

Input: Image
Output: Matrix

```
1 for all grid points  $(i, j)$  do
2   | if  $(i, j) \in R$  then
3   |   |  $A^0[i, j] = 1$ 
4   | end
5   | if  $(i, j) \notin R$  then
6   |   |  $A^0[i, j] = -1$ 
7   | end
8 end
9  $n = 0$ 
10 while  $n < \text{stopping condition}$  do
11   | for all grid points  $(i, j)$  do
12   |   | Update  $F_{i,j}^n$ 
13   | end
14   | for all grid points  $(i, j)$  do
15   |   |  $K_{i,j}^n = \kappa$ 
16   |   |  $A^{n+1} =$ 
17   |   |  $A^n + \Delta t(-[\max(F_{i,j}^n, 0)\nabla^+ + \min(F_{i,j}^n, 0)\nabla^-] + \lambda_{\text{length}}K_{i,j}^n((D_{i,j}^{0x})^2 + (D_{i,j}^{0y})^2)^{1/2})$ 
18   | end
19   |  $n = n + 1$ 
20 end
21 return  $A^n$ 
```

4.3 Algorithm Time Step

When implementing the discretization as described above, it was important to choose a time step small enough to ensure the curve evolves in a stable manner. Originally, the time step was chosen experimentally to be an arbitrarily small number such that the curve evolution was stable. However, under these conditions the curve evolution was very slow. Moving forward, the Courant-Friedrichs-Lowy (CFL) condition was considered [20]. This is a necessary condition for convergence while solving partial differential equations numerically. The condition considers the idea that the curve cannot advance or recede more than one pixel per time step. The CFL condition states that for a time-marching computer simulation to produce correct results, the time step must be less than a given time, Δt , where

$$\Delta t \leq \frac{1}{\max_{\gamma} F^n}.$$

Computationally, it is difficult to determine γ , therefore, F^n was maximised over all i, j . In order to ensure stability, Δt was defined as follows

$$\Delta t = \frac{0.75}{\max_{i,j} F^n}.$$

4.4 Stopping Condition

A stopping condition was introduced to stop the region tracking algorithm after defined conditions of convergence had been met. This was defined by the number of points, (i, j) , where

$A_{i,j}^{n-1} A_{i,j}^n < 0$. A count of these points was kept for each iteration of the algorithm. Once this count dropped below a threshold for a pre-defined number of iterations, the region was deemed convergent and the algorithm stopped. The threshold and the number of iterations are determined experimentally depending on the image type.

4.5 Image Definition

To derive the functionals used to describe the region of interest, each image was defined as an intensity map, $I(x, y)$. $I(x, y)$ is defined at each point of a two-dimensional image domain $\Omega \subset \mathbb{R}^2$. The range of $I(x, y)$ is dependent on the type of image used. For black and white images, each pixel in the image takes a value of either 0 or 255, more precisely, the mapping of $I(x, y)$ is as follows: $I : \Omega \rightarrow \{0, 255\}$. If the image is grey-scale, then the range is from 0 to 255 inclusive. In other words, for a grey-scale image, $I : \Omega \rightarrow \{0, 1, \dots, 255\}$. For images with colour, $I(x, y)$ is a vector valued function mapping into some colour space, normally RGB (Red, Green, Blue). Only grey-scale images were considered in this project, thus, each pixel takes a value in the range $[0, 255]$. In this range, 0 is indicative of an absolute black pixel and 255, an absolute white pixel.

4.6 Functional Evaluation Metric

A testing dataset and evaluation metric were chosen to quantitatively measure the performance of functionals in the following section. The data set chosen was a collection of GFP-GOWT1 mouse stem cell images, taken from the Cell Tracking Challenge database [21]. The evaluation process considers the algorithm's performance in tracking the cell between the two test frames, 4.1a and 4.1b.

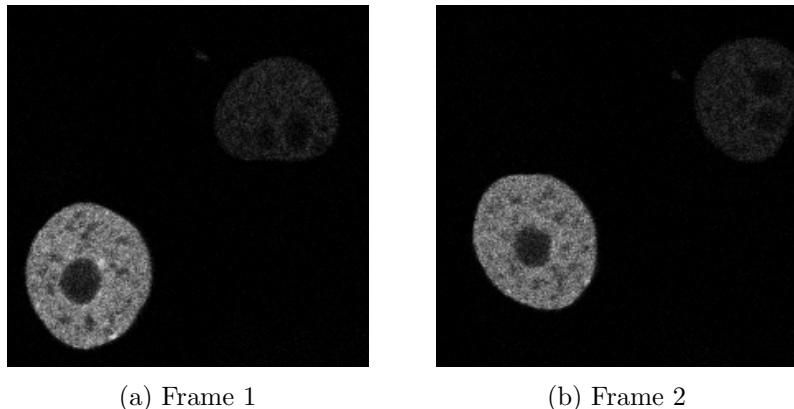


Figure 4.1: GFP-GOWT1 mouse stem cell baseline image set [21]

Using the region tracking algorithm in a medical application requires high levels of sophistication and extremely accurate tracking. The evaluation methodology will follow that of the Cell Tracking Challenge. The results of the algorithm, R , will be evaluated against the ground truth, S , using the Jaccard similarity index (J). Where the Jaccard similarity index is

$$J(S, R) = \frac{|R \cap S|}{|R \cup S|}, \quad (4.1)$$

with $J \in [0, 1]$. A superior performance maximises the selection of desired region by the algorithm and minimises the selection of undesired region. An index of 1 indicates a perfect match (perfect performance) while an index of 0 indicates no match. The best 3 results achieved

on this data set in the Cell Tracking Challenge had Jaccard similarity indices of: 0.927, 0.921 and 0.894 [3]. The benchmark for our algorithm will be the average of these three results, 0.914, as an average of the Jaccard similarity performance across all frames in the data set.

5 Design Solution

The proposed design solution is a computer vision algorithm, with the input being an identified region within an image, and the output being the tracking of the identified region through a sequence of images. In the context of the application, a stem cell will be identified and tracked through a sequence of microscopic images.

5.1 Invariant Properties

Certain assumptions were made regarding the time-invariant properties of the cell images sequences. These invariant properties were used in the selection of appropriate functionals to form the region tracking functional.

The data set images are in greyscale, with the cells moving on a background of invariant colour, typically black or grey. Thus, it was assumed that the desired region, or the cell, is invariant under intensity of the region. It was also assumed that the cells have reached maturity, indicating that their shape and area are not changing. Thus, shape and area of the cells were considered time-invariant. While there may be slight changes in these properties between the start and end of the image sequence, the changes between frames are negligible. Velocity vectors are bound to the x and y co-ordinate directions as the cells are tracked in a 2-dimensional plane. This reduces the complexity of the tracking as well as the region of consideration required around each original region.

In summary, the time-invariant properties of the cell images are as listed:

- Intensity
- Mean intensity
- Intensity Variance
- Area
- Shape

5.2 Functional Design

The following sections discuss the design of specific functionals that correspond to the time-invariant properties identified above. The final region tracking functional will be a linear combination of the designed functionals.

5.2.1 Length Functional

The length functional is given by (5.1)

$$E_{\text{length}}[\gamma] = \lambda_{\text{length}} \int_0^1 \|\gamma(s)\| ds \quad (5.1)$$

where λ_{length} is a weighting term. The length functional is minimised when the length of the curve is zero, as shown in Figure 3.4. The length functional will contribute to the reduction of the area of the region enclosed, and reduce the effect of noise on the performance of the algorithm. Following the work of Dr. Abdol-Reza Mansouri, an initial value of 10 was given to λ_{length} [10].

5.2.2 Intensity Functional

As discussed in Section 5.1, the intensity of the cell is assumed to remain invariant. This is due to the fact that the luminous intensity of the cell, and the background intensity, are constant. Each pixel intensity is individually compared against the mean value of the region it lies within. The mean values are calculated for the original region, where a is the average intensity of the pixels inside of the original region and b , is the average intensity of the pixels outside of the original region. The intensity functional is given by (5.2).

$$E_{\text{intensity}}[\gamma] = \lambda_{\text{intensity}} \left(\iint_{R_\gamma} (I_1(x, y) - a)^2 dx dy + \iint_{R_\gamma^c} (I_1(x, y) - b)^2 dx dy \right) \quad (5.2)$$

The term $\lambda_{\text{intensity}}$ is the weighting term for the functional. The discretization of the intensity functional, implemented in the algorithm, is given by (5.3).

$$F_{\text{intensity}} = (I_1(x, y) - a)^2 + (I_1(x, y) - b)^2 \quad (5.3)$$

As each pixel value is considered individually, it is expected that this functional will perform well in segmentation with black and white images.

Testing the functional, with $\lambda_{\text{intensity}} = 1$ and $\lambda_{\text{length}} = 10$, on the test frames gave the results shown in Figure 5.1a and Figure 5.1b in 44 iterations. These results have a Jaccard similarity index score of 0.83556. The functional tightly fits the outside of the cell, but removes clusters of black pixels within the cell due to the individual consideration of pixels.

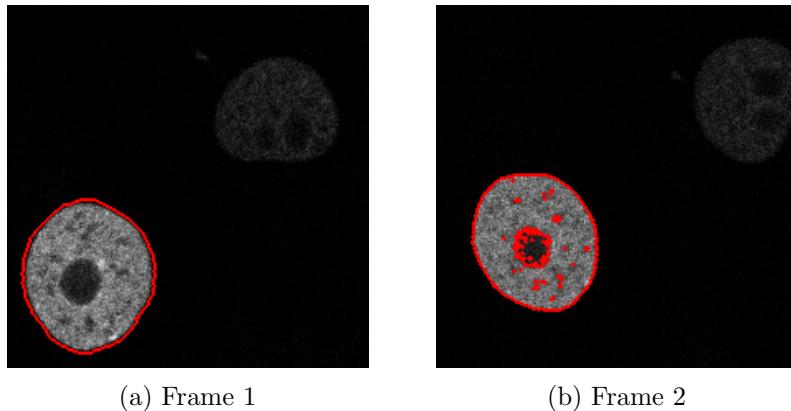


Figure 5.1: Testing the intensity functional

The intensity functional under-performs the Jaccard similarity index performance benchmark on a single frame. Qualitatively, the functional is minimized by removing all the black pixels from the region that are part of the desired region of the cell. However, the intensity functional converges to a result quickly, returning a minimizing region in 44 iterations.

5.2.3 Mean Intensity Functional

In attempt to better describe the distribution of pixels within the region, a mean intensity functional was created. In each iteration, this functional compares the average of pixels within the region against the average of the pixels in the original region. The mean intensity functional

is given by the following (5.4).

$$E_{\text{mean intensity}}[\gamma] = \lambda_{\text{mean}} \left(\left(\frac{\iint_{R_\gamma} I_1(x, y) dx dy}{\iint_{R_\gamma} dx dy} - a \right)^2 + \left(\frac{\iint_{R_\gamma^c} I_1(x, y) dx dy}{\iint_{R_\gamma^c} dx dy} - b \right)^2 \right) \quad (5.4)$$

The term λ_{mean} is the weighting term for the functional. The discretization of the mean intensity functional is as described in (5.5).

$$\begin{aligned} F_{\text{mean intensity}} &= 2 \left(\frac{\iint_{R_\gamma} I_1(x, y) dx dy}{\iint_{R_\gamma} dx dy} - a \right) \frac{I_1(x, y) \iint_{R_\gamma} dx dy - \iint_{R_\gamma} I_1(x, y) dx dy}{(\iint_{R_\gamma} dx dy)^2} \\ &\quad - 2 \left(\frac{\iint_{R_\gamma^c} I_1(x, y) dx dy}{\iint_{R_\gamma^c} dx dy} - b \right) \frac{I_1(x, y) \iint_{R_\gamma^c} dx dy - \iint_{R_\gamma^c} I_1(x, y) dx dy}{(\iint_{R_\gamma^c} dx dy)^2} \end{aligned} \quad (5.5)$$

Testing the mean intensity functional on the test frames with $\lambda_{\text{mean}} = 1$, gave the results shown in Figure 5.2a and Figure 5.2b in 800 iterations. These results have a Jaccard similarity index score of 0.93253.

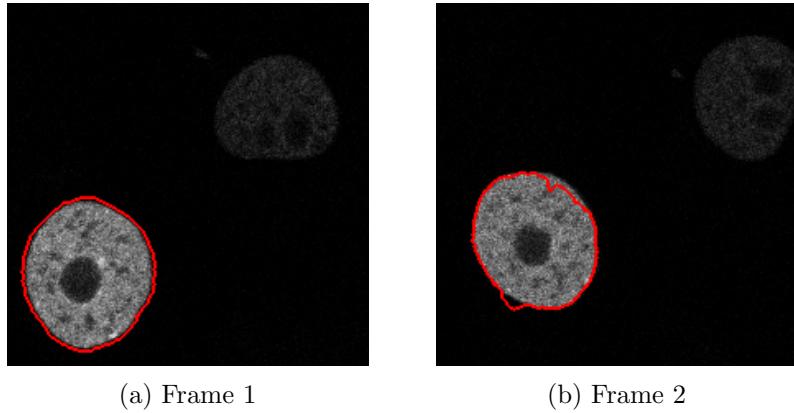


Figure 5.2: Testing mean intensity functional

The mean intensity functional outperforms the benchmark performance on this frame. Additionally, the problem of removing the center of the cell from the tracked region seems to have been solved. Qualitatively, the region could still be fit better, as there is an odd discrepancy in the shape of the outline of the cell. The mean intensity functional takes significantly longer to converge than the intensity functional, taking a total of 800 iterations. This indicates that there is information yet to be captured by the functionals.

5.2.4 Area Functional

To capture the invariance of the cell's area, the region of each iteration is compared to the area of the original region, A . The quadratic cost functional is given by (5.6)

$$E_{\text{area}}[\gamma] = \lambda_{\text{area}} \left(\iint_{R_\gamma} dx dy - A \right)^2 \quad (5.6)$$

where λ_{area} is the weighting term. The discretization of the area functional is described in (5.7).

$$F_{\text{area}} = 2 \left(\iint_{R_\gamma} dx dy - A \right) \quad (5.7)$$

Testing the area and mean intensity functionals together on the test frames, with $\lambda_{\text{mean}} = 1$ and $\lambda_{\text{area}} = 0.0001$, gave the results shown in Figures 5.3a and 5.3b in 950 iterations. These results have a Jaccard similarity index score of 0.90940.

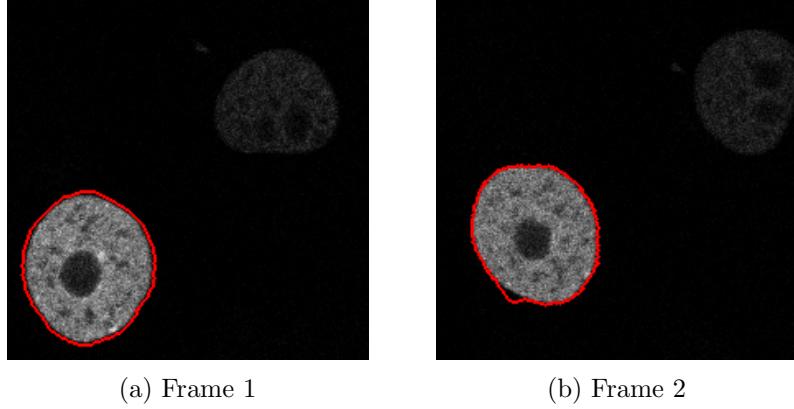


Figure 5.3: Testing combined mean intensity and area functionals

The area functional performs below the performance benchmark on this frame, and takes an extra 150 iterations to converge to a minimizer when compared to the mean intensity functional alone. Qualitatively, the area functional fits the region better, with only one major deviance on the boundary.

5.2.5 Intensity Variance Functional

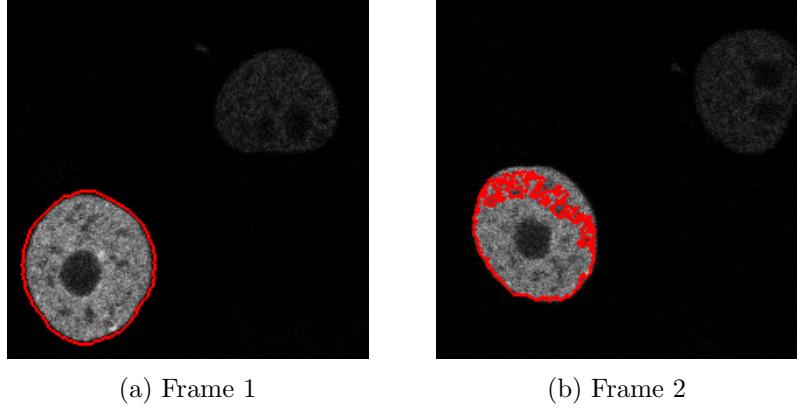
To capture the invariance of pixel intensities within the original region, an intensity variance functional was created. This functional is especially useful in the process of correctly characterising a region. The functional is given by (5.8)

$$E_{\text{variance}}[\gamma] = \lambda_{\text{variance}} \left(\frac{\iint_{R_\gamma} I_1(x, y)^2 dx dy}{\iint_{R_\gamma} dx dy} - \left(\frac{\iint_{R_\gamma} I_1(x, y) dx dy}{\iint_{R_\gamma} dx dy} \right)^2 - a \right)^2 \quad (5.8)$$

where $\lambda_{\text{variance}}$ is the weighting term. The discretization of the variance functional is described in (5.9)

$$\begin{aligned} F_{\text{variance}}[\gamma] &= 2 \left(\frac{\iint_{R_\gamma} I_1(x, y)^2 dx dy}{\iint_{R_\gamma} dx dy} - \left(\frac{\iint_{R_\gamma} I_1(x, y) dx dy}{\iint_{R_\gamma} dx dy} \right)^2 - a \right) \\ &\quad \times \left(\frac{I_1(x, y)^2 \iint_{R_\gamma} dx dy - \iint_{R_\gamma} I_1(x, y)^2 dx dy}{(\iint_{R_\gamma} dx dy)^2} \right. \\ &\quad \left. - 2 \frac{\iint_{R_\gamma} I_1(x, y)^2 dx dy I_1(x, y) \iint_{R_\gamma} dx dy - \iint_{R_\gamma} I_1(x, y) dx dy}{(\iint_{R_\gamma} dx dy)^3} \right). \end{aligned} \quad (5.9)$$

Testing the mean intensity and intensity variance functionals together on the test frames, with $\lambda_{\text{mean}} = 1$ and $\lambda_{\text{variance}} = 0.001$, gave the results shown in Figures 5.4a and 5.4b in 1175 iterations. These results have a Jaccard similarity index score of 0.88706.



(a) Frame 1

(b) Frame 2

Figure 5.4: Testing combined mean intensity and variance functionals

The intensity variance functional performs below the performance benchmark, qualitatively does not track the region well and takes a large amount of iterations to converge to a minimizer. Adding an area functional to this linear combination may help boost performance.

5.3 Region Tracking Functional

The final functional is a linear combination of the individual functional terms discussed in the above sections. This defines the final level set PDE as the sum of individual level set equations.

We have

$$E[\gamma] = \sum_{i=1}^N E_i[\gamma], \quad (5.10)$$

and

$$\frac{\partial u}{\partial t} = - \sum_{i=1}^N \lambda_i F_i \|\nabla u\|$$

where each i is the functional name as defined in the sections above, and N is the number of individual functionals chosen for inclusion in the final functional. The functional must be designed such that its global minimizer is the curve $\gamma_1 \in C$.

A major factor of functional design is determining which individual functionals should be included in the final functional and their corresponding weightings, λ_i . These λ_i are determined experimentally through trial and error and vary depending on the data set of images. The weightings are used to determine which factors, defined by the functionals, are more important in defining a minimizer. The performance of the functionals on the test data set gives a point from which to begin this trial and error process. Each of the tested functionals captures valuable information about the selected region and thus all five are included in the implementation. The initial weightings, determined by the performance of the functionals on the test set, were $\lambda_{\text{length}} = 10$, $\lambda_{\text{intensity}} = 1$, $\lambda_{\text{mean}} = 1$, $\lambda_{\text{area}} = 1$, and $\lambda_{\text{variance}} = 0.0001$.

The tuning of λ_i for each functional includes iterative design, and design trade offs. Each λ_i is a function of the others, thus, when one weighting is changed slightly, each of the other weightings must also be altered to ensure the functional converges as desired. Each linear combination of individual functionals is a trade off between speed of convergence of the algorithm and accuracy

of performance. Speed of convergence is important to consider in the functional weighting as both computation power and time are limited in the scope of this project.

6 Implementation and Evaluation of Results

To begin implementation, the functional was run on a subsection of 2 images from the GFP-GOWT1 mouse stem cell image data set [21]. This was done with the objective of outperforming the performance benchmark of an average Jaccard similarity index score of 0.914, as discussed in Section 4.6. The results of the first trial are shown in Figure 6.1, and the Jaccard similarity index is below the target value.

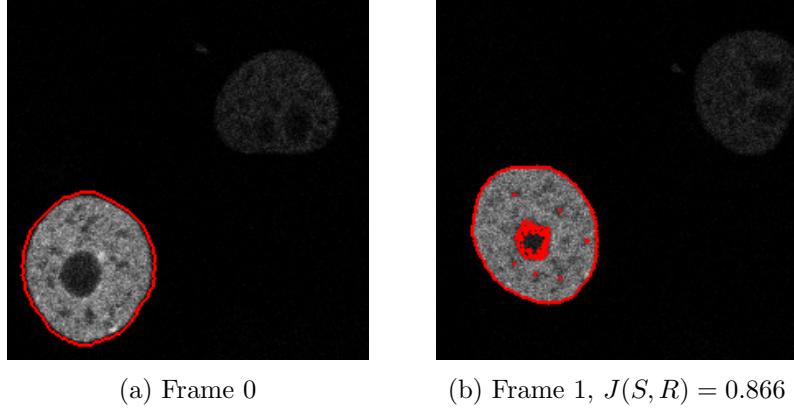


Figure 6.1: Tracking performance

Although the functional precisely outlines the boundary of the cell, there is undesired region selected within the cell. After seeing many similar outcomes through experimental testing, it was concluded that the intensity functional failed to differentiate between the nucleus of the cell, and the background of the image. To combat this issue, different image pre-processing methods were considered.

First, image filtering was used in attempt to enhance the images in the data set and increase the difference in pixel values between the cell nucleus and image background. A lightness filter was applied to the images in the data set to increase contrast in intensity between the cell pixels and the pixels in the background of the image. The lightness filter applies a linear multiplier to the input image, making it appear brighter. The results of this filtering can be seen in Figure 6.2a (before filtering) and 6.2b (after filtering).

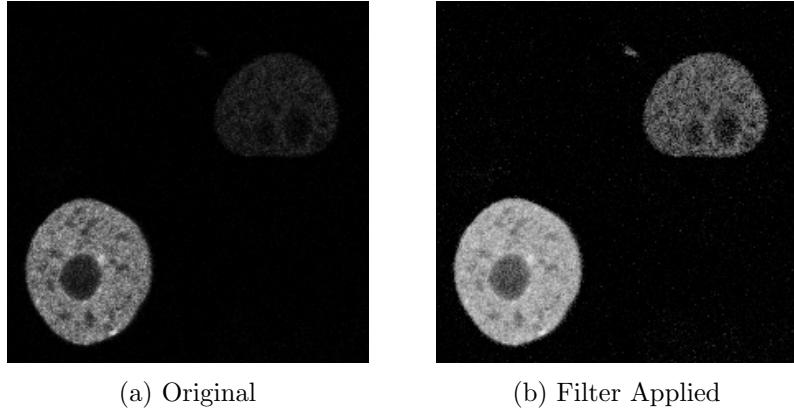


Figure 6.2: Image filtering results

The results of the filtering are summarized in Table 6.1. The difference between the average

pixel intensity of the cell pixels and the background pixels was increased by 38.98, from 95.45 before filtering, to 137.77 after filtering.

Table 6.1: Results of filtering on the cell images

| | Before Filtering | After Filtering |
|-------------------------------|------------------|-----------------|
| Cell average pixel intensity | 98.79 | 146.68 |
| Outer average pixel intensity | 3.34 | 8.91 |
| Difference | 95.45 | 137.77 |

The results of lightness filtering on tracking performance are shown in Figure 6.3. Applying a lightness filter in pre-processing increased the Jaccard similarity index performance over one frame from 0.866 (below the performance benchmark) to 0.930, which is above the performance benchmark.

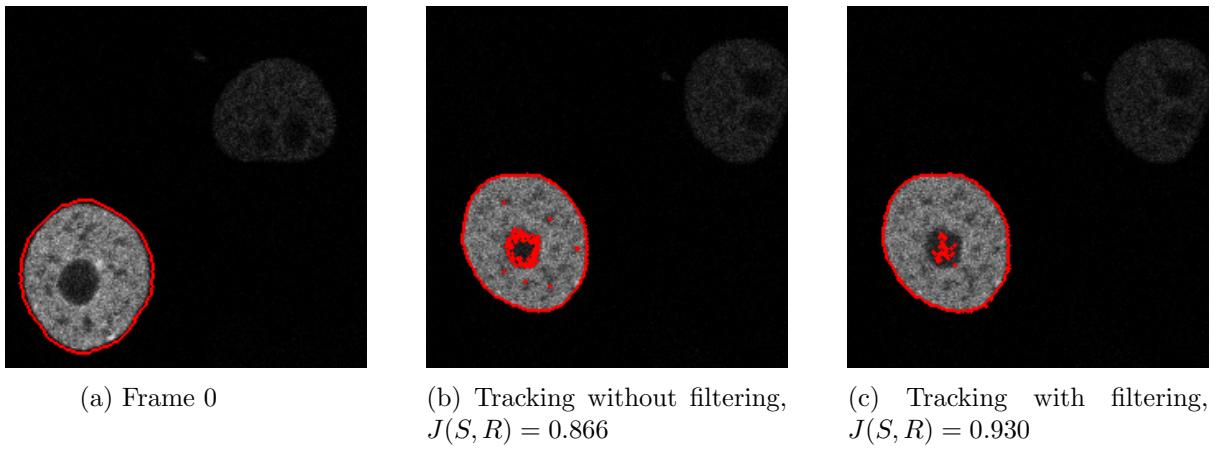


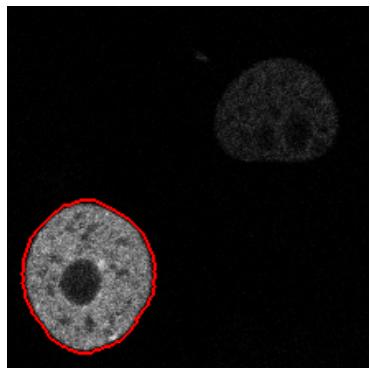
Figure 6.3: Effects of filtering on tracking performance

The functional was then run on a subsection of 9 images from the GFP-GOWT1 mouse stem cell image data set, after pre-processing the images with a lightness filter [21]. Once again, this was done with the objective of outperforming the performance benchmark of an average Jaccard similarity index score of 0.914 over the 9 frames, as discussed in Section 4.6.

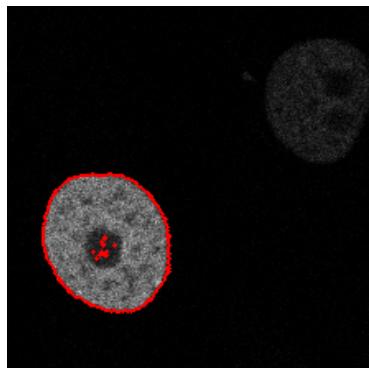
As discussed in Section 5.3, a set of initial weightings, determined by the performance of the functionals on the test set, were set to $\lambda_{\text{length}} = 10$, $\lambda_{\text{intensity}} = 1$, $\lambda_{\text{mean}} = 1$, $\lambda_{\text{area}} = 1$, and $\lambda_{\text{variance}} = 0.0001$. An iterative process of running the algorithm, analysing the results and tuning the functional weightings accordingly was performed. With each choice of weightings the Jaccard similarity index scores, speed of convergence and qualitative improvements (such as not selecting undesired region or better edge tracking) were considered. The functional weightings that produced the best results were:

$$\lambda_{\text{length}} = 10, \lambda_{\text{intensity}} = 0.5, \lambda_{\text{mean}} = 2.5, \lambda_{\text{area}} = 1.5, \text{ and } \lambda_{\text{variance}} = 0.001$$

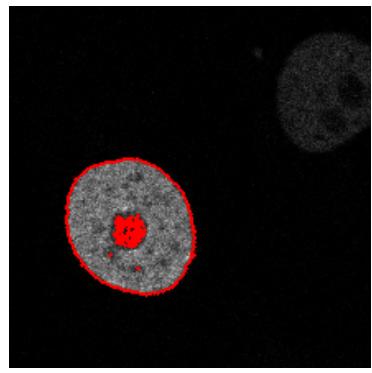
The performance of the algorithm on the testing data set can be seen in Figure 6.4. The average Jaccard similarity index of these frames is 0.886, below the benchmark performance of the Cell Tracking Challenge. Qualitatively, we can see that the outer boundary of the cell is tracked quite accurately in each frame. However, the nucleus of the cell was removed from the cells, likely due to its similarity with the black background.



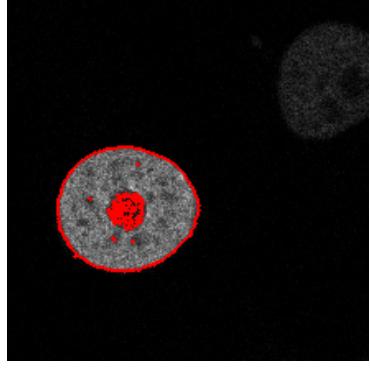
(a) Frame 0



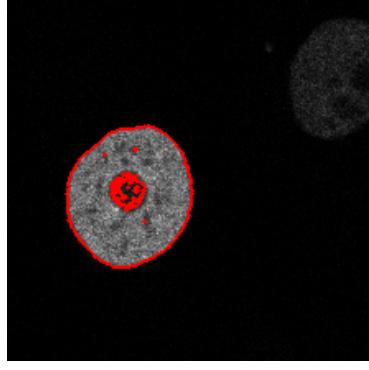
(b) Frame 1, $J(S, R) = 0.939$



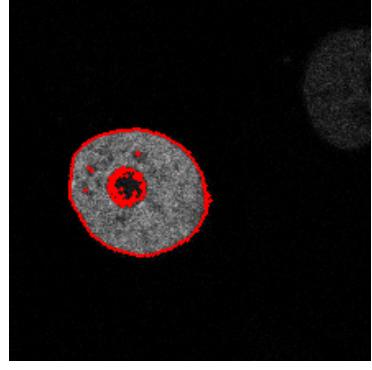
(c) Frame 2, $J(S, R) = 0.904$



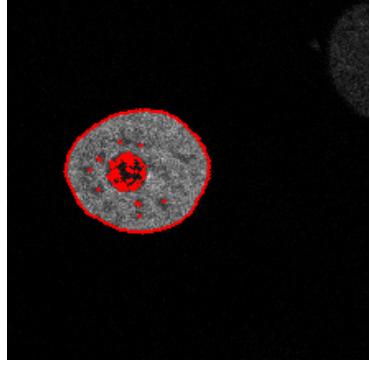
(d) Frame 3, $J(S, R) = 0.886$



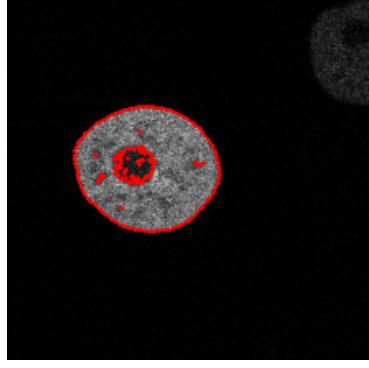
(e) Frame 4, $J(S, R) = 0.876$



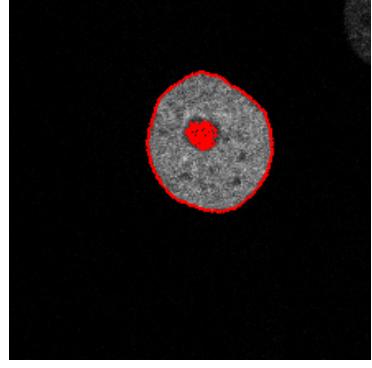
(f) Frame 5, $J(S, R) = 0.871$



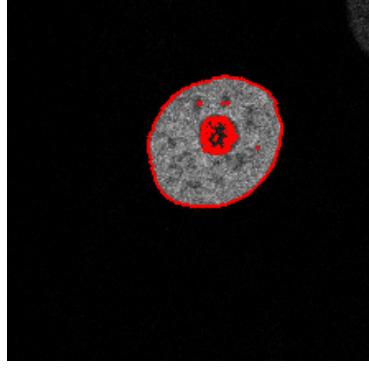
(g) Frame 6, $J(S, R) = 0.870$



(h) Frame 7, $J(S, R) = 0.855$



(i) Frame 8, $J(S, R) = 0.902$



(j) Frame 9, $J(S, R) = 0.868$

Figure 6.4: Tracking on mouse stem cell data set

Next, the functional was run on a subsection of 9 images from the NYU Center for Genomics and Systems Biology mouse embryo cell data set [22]. One of the images from the mouse embryo cell data set is shown in Figure 6.5. The benchmark performance for this data set is established in a paper by Cicconet et al [22]. Using automated region tracking, the team achieved a Jaccard similarity index score of 0.95, averaged across the data set.

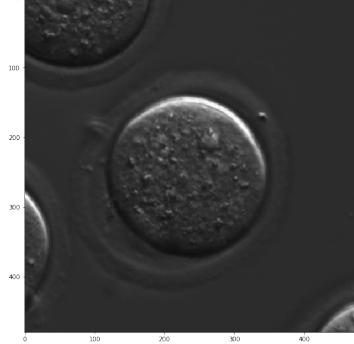


Figure 6.5: Mouse embryo cell image frame

Using a similar process to that described earlier in this section, the individual functional weights were tuned for the new data set, resulting in the following values:

$$\lambda_{\text{length}} = 10, \lambda_{\text{intensity}} = 0.5, \lambda_{\text{mean}} = 2.5, \lambda_{\text{area}} = 1.5, \text{ and } \lambda_{\text{variance}} = 0.1$$

The objective of this trial was to observe the effects of reducing the variation between each frame on the accuracy of the tracking algorithm. To test this characteristic, the time step between frames was decreased. The tracking results for this trial are shown in Figure 6.6 on the following page. The average Jaccard similarity index for these frames is 0.953. This is above the benchmark performance established by Cicconet et al., indicating that the algorithm can produce more optimal results by reducing the time steps and resultingly the variance between frames.

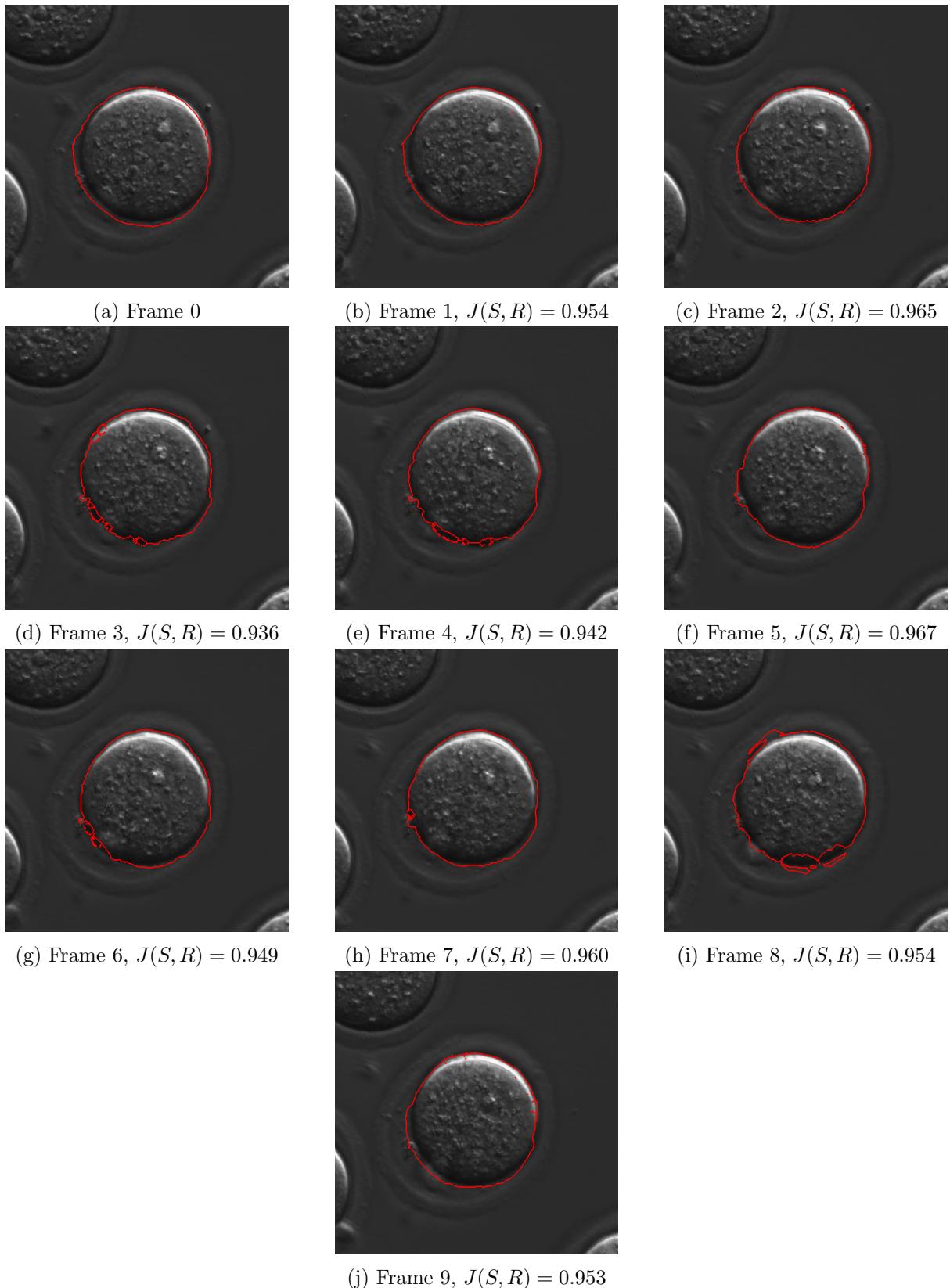


Figure 6.6: Tracking on mouse embryo cell data set

In attempt to further optimize the results of the tracking algorithm, different filters were tested

on the data set: a dilation filter and an adaptive thresholding filter.

Dilation and erosion are two fundamental morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels from the boundary. The algorithm was tested using data sets with dilation filters. It was expected that the filter would improve results since the intensity of the boundary pixels are not constant between frames in the data set. The effect of dilation filtering is shown in Figures 6.7a and 6.7b.

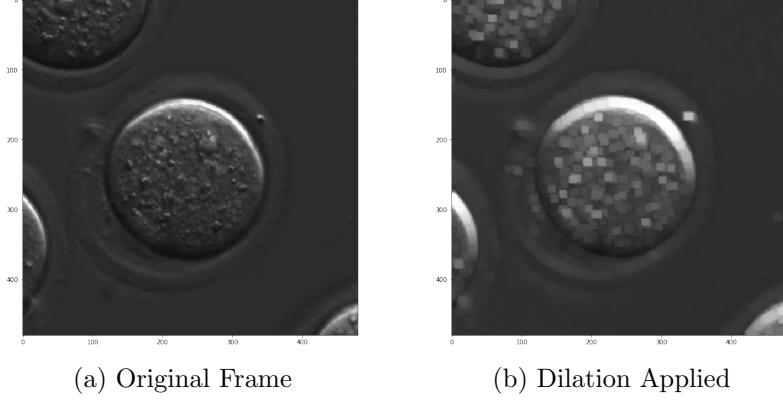


Figure 6.7: Dilation filtering results

The effect of the dilation filter on tracking accuracy is shown in Figures 6.8a, 6.8b and 6.8c. It was found that dilation reduced the Jaccard similarity index and was therefore not used in the implementation.

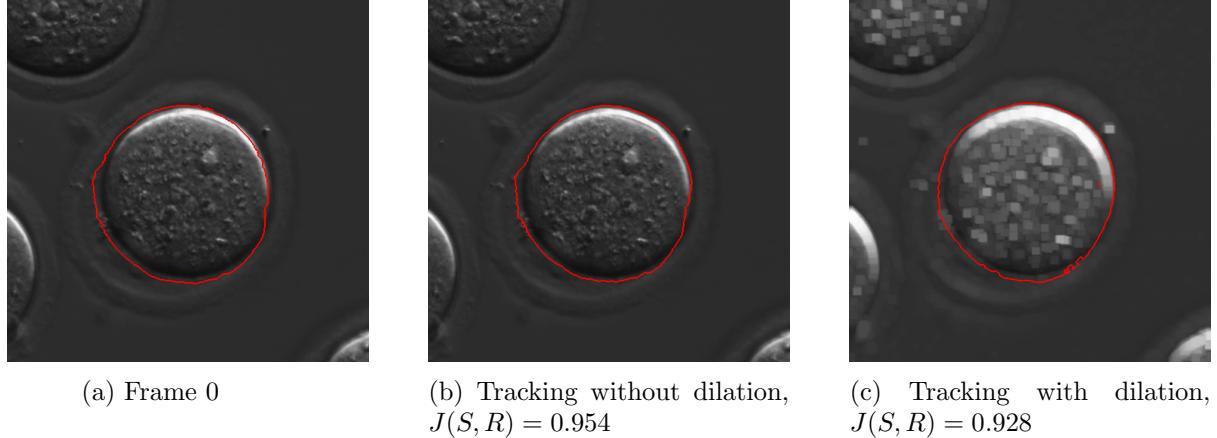
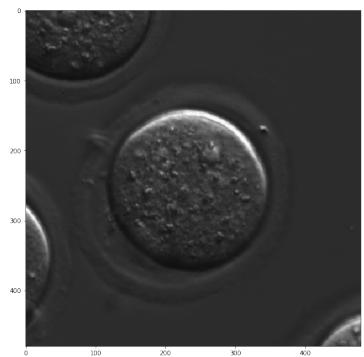
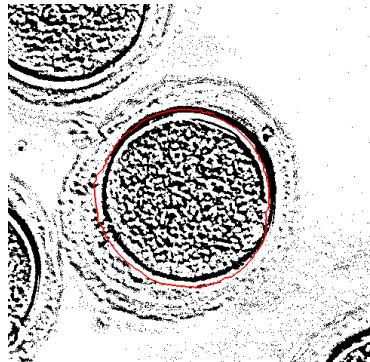


Figure 6.8: Effects of dilation filtering on tracking performance

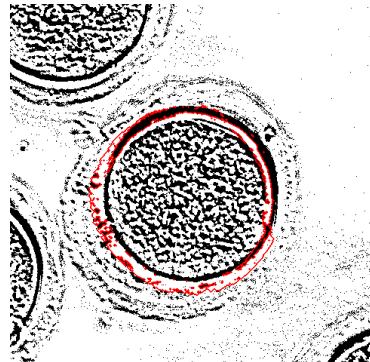
Thresholding is a pre-processing technique that involves picking a threshold value, setting all pixels with intensity less than that value to zero, and otherwise making the intensity equal to the threshold value. Adaptive thresholding computes the best threshold value for different areas of the image rather than using a global threshold intensity. It was determined through trial and error that a threshold value of 40 was optimal for the boundary. The effect of adaptive thresholding are shown in Figures 6.9a, 6.9b and 6.9c.



(a) Original Frame



(b) Thresholding applied



(c) Tracking region

Figure 6.9: Adaptive thresholding filtering results

Results indicate that this filtering technique does not improve tracking accuracy. This is most likely because the algorithm was not fine tuned to perform with images that are binary black and white. Due to the project timeline, no further optimization processes were tested.

7 Societal, Ethical, Environmental and Economic Impacts

7.1 Societal Impact

One of the main motivations for automated region tracking is to accelerate the process of medical image analysis. Advancements in medical region tracking will directly benefit professionals' ability to understand and diagnose numerous diseases. When considering this engineering initiative applied specifically to stem cell research, the impact on society becomes even more evident. Many areas of stem cell research are still very foreign. Professionals believe that continued study of stem cells will lead to new discoveries in regenerative medicine and improve patient outcomes. The safety, efficacy and quality of pursuing stem cell research in Canada is the responsibility of *Health Canada* under the authority of the federal *Food and Drugs Act* along with its accompanying regulations [23], [24].

One avenue of stem cell research with huge potential is for the treatment of Alzheimer's disease. There are currently 5.8 million Americans living with Alzheimer's Disease, and this number is expected to rise to nearly 14 million by 2050 [6]. A greater understanding of the disease could also aid in early diagnoses. Among citizens at age 70, 61% of people with Alzheimer's are expected to die before the age of 80, compared to 30% of people without Alzheimer's [6]. Treatments developed as a result of stem cell research have the potential to improve the lives of patients who live with Alzheimer's and help prevent its progression to more advanced stages.

7.2 Ethics

The implementation of stem cell research invokes a number of ethical considerations. Stem cells used in research are either adult stem cells or embryonic stem cells. In the case of adult stem cells, professionals are ethically inclined to receive consent from the individual(s) whose stem cell data will be used.

When working with embryonic stem cells, the question of the "sanctity of life" must be considered. The "sanctity of life" ethic views an embryo as a human – implying that these embryos possess human rights. Because embryos would be destroyed during research processes some consider embryonic stem cell research to be unethical. In Canada, stem cell research was legalized in 2004 under the stipulation that embryos used in research are a surplus from fertility clinics. The *Assisted Human Reproduction Act* legislates any research done using embryonic stem cells and regulates the collection of stem cells [24]. Today, Canada continues to ban the creation of embryos for research and therapeutic cloning purposes [23].

7.3 Environmental Impact

The ethics of stem cell research must be considered from an environmental standpoint as well. Stem cell research commonly uses animal testing to study the biological response to cell transplants. A recent study involved injecting stem cells into the brain of a mouse and monitoring the cells as they multiplied and integrated with existing brain tissue [25]. These processes receive controversial attention. In addition, in all cases where animal cells are used as subjects of a study, the cells must be obtained humanely. Due to the availability of data, animal stem cell images were used for the thesis.⁴

⁴We evaluated the source of the animal stem cell images, and gave consideration to the ethics discussed.

Success in stem cell research will impact the field of toxicology. Researchers at the *Chinese Academy of Sciences* and *The University of California at San Diego* have detailed the use of embryonic stem cells to gauge the neurotoxicity effects of the environmental pollutant, Biphenol A (BPA), on human health [26]. The results of the study revealed that BPA may alter embryonic development in vivo. This shows that embryonic stem cells could serve as a model to evaluate the physiological effects of pollutants effectively and cost-effectively. The use of region tracking algorithms will help automate and further this field of research in order to develop a deeper understanding of environmental pollutants and their effects on humans. This research in toxicology using stem cells has the potential to influence environmental policies which aim to reduce the release of pollutants found to have negative effects.

Conversely, the advancement of stem cell research and the applications in treatments will increase the demand for pharmaceuticals which will directly add to environmental pollution. The pharmaceutical industry is increasingly recognised as a threat to the environment as one of the largest contributors to global warming with emission intensity about 55% higher than that of the automotive industry [27]. The demand for pharmaceuticals is vast with 100,000 tonnes of pharmaceutical products consumed globally every year. In addition to emissions created during production, over 30% of pharmaceutical products are returned as an active substance to the environment. These potent chemicals are then returned to drinking water, food systems and ecosystems [28]. In the United States, the *Office of Pollution Prevention and Toxics* manages programs under the *Toxic Substances Control Act* and the *Pollution Prevention Act*. Under these laws, new and existing chemicals and their risks are evaluated and ways to prevent or reduce pollution are considered before the chemicals are integrated into the environment [29].

7.4 Economic Impact

Automated region tracking when applied to stem cell research has the potential to aid in the development of new drugs and medical procedures which will introduce profit opportunities for both the pharmaceutical and healthcare industries as well as reduce healthcare costs for individuals. Specifically, stem cell research is expected to be used in Alzheimer's treatment in the future. The total cost of health care and long-term care services for Alzheimer's patients is projected to cost the American economy \$290 billion this year. This number is projected to rise to \$1.1 trillion by 2050 [6]. The improved treatments and procedures derived from successful stem cell research have the potential to reduce the number of citizens living with Alzheimer's as well as the cost of their treatment, which would drastically reduce the 2050 projected costs.

In addition, the ability to track a sequence of images of stem cells using a tracking algorithm would greatly improve the efficiency of tracking stem cells in comparison to conventional methods. Large databases of valuable information could be created with less employees and in a shorter amount of time. More efficient data processing, made possible by automated region tracking, would also aid in stem cell analysis in other professions such as for cell biologists. Additionally, the adaptability of the proposed tracking algorithm lends itself to being applied to many other medical imaging processes, reducing capital expenditures for hospitals and clinics. At the University of Malaya Medical Centre, it was found that the anatomic pathology laboratory cost the hospital US \$62 per case and expenditures totaled US \$1.20 million in 2015 [30]. Region tracking will accelerate the process at which scans, and other medical images, can be analyzed and minimize the role of pathologists and radiologists in the analysis process. This will reduce the total cost of both pathology and radiology in a hospital setting.

Although the proposed solution could bring cost savings to hospitals and clinics, it must be considered that appropriate technology is required to run the algorithm. The level of sophistication of the perfected algorithm would require powerful computers capable of processing large data sets in an efficient manner. This could increase computing costs associated with research.

8 Conclusions and Future Work

When tested on the GFP-GOWT1 mouse stem cell image data set it was found that the algorithm can initially track very accurately, but becomes less accurate as more frames are tracked. While the target Jaccard similarity index of 0.914 is achieved on the first frame, it is not consistently reached over nine frames [3]. The algorithm proved to be more accurate on the NYU Center for Genomics and Systems Biology data set and achieved the baseline Jaccard similarity index performance over nine frames. This indicates that while the algorithm has the potential to track at a high level of accuracy, it is not consistent.

The imperfect performance of the algorithm on the mouse stem cell data set indicates that the invariant properties of the images were not fully captured. This caused of the removal of specified cell's nucleus from the region of interest in each frame. In a medical or research application, this would be unacceptable as the removal of the nucleus drastically alters any data generated from the tracked cell images. In order to improve the algorithm and prevent the removal of the dark cell nucleus, additional functionals should be implemented which better capture the distribution of pixel intensities in the region. One interesting function is the Kullback-Leibler Divergence functional. The Kullback-Leibler Divergence is a measure of how different one probability distribution is from a reference probability distribution, and could thus help the algorithm to converge closer to the region of interest.

One of the weaknesses of the algorithm is the slow run-time as it must iterate through large arrays to calculate functional values at each step. Despite the algorithm being run on small images (400×400 pixels), each frame takes over 30 minutes to run. In an application setting, many data sets of higher quality images would need to be analysed. This is currently not feasible as increasing the image size would further delay the run-time. One method of increasing processing speed is to reduce the region of interest to a smaller scope under the assumption that cells will only move a certain distance between frames. This entails cropping images down to a smaller radius around the cells and reducing the pixels of interest to those within the small radius around the original cell of interest.

A major assumption made in this project was that the examined cells are mature. This greatly limits the scope of the application. To properly evaluate the performance of this approach, the tracking algorithm should be tested on a wide range of cell types, such as the array of data sets provided by the Cell Tracking Challenge [3]. This was not feasible in this report due to time constraints.

The above discussion introduces just a few of the steps that could be taken to make the automated region tracking algorithm more consistent, accurate and efficient. With more time and resources, the current algorithm could be further developed to match the quality of the top algorithms that have been submitted to the Cell Tracking Challenge [3].

A Important Theorems

Theorem A.1 (Chain Rule [31]). *If each component of the curve $\mathbf{x} = \mathbf{x}(t)$ is differentiable at $t = t_0$, and that $f = f(\mathbf{x})$ is differentiable at $\mathbf{x}_0 = \mathbf{x}(t_0)$, then*

$$\frac{df(\mathbf{x}(t))}{dt}\Big|_{t=t_0} = \nabla f(\mathbf{x}_0) \dot{\mathbf{x}}(t_0) = \sum_{k=1}^n \frac{\partial f}{\partial x_k}\Big|_{\mathbf{x}=\mathbf{x}_0} \frac{dx_k}{dt}\Big|_{t=t_0}.$$

Theorem A.2 (Jordan Curve Theorem [32]). *Any simple closed contour separates the plane into two domains each having the curve as its boundary. One of these domains, called the interior, is bounded; the other, called the exterior is unbounded.*

Theorem A.3 (Gage and Hamilton [7]). *If γ is a convex curve embedded in the plane \mathbb{R}^2 , the heat equation shrinks the curve γ to a point. The curve remains convex and becomes circular as it shrinks, in the sense that*

1. *the ratio of the inscribed radius to the circumscribed radius approaches 1 and the ratio of the maximum curvature to the minimum curvature approaches 1;*
2. *the higher order derivatives of the curvature converge to 0 uniformly.*

Theorem A.4 (Grayson [8]). *Let γ be a smooth embedded curve in the plane. Then $\gamma : [0, 1] \times [0, T] \rightarrow \mathbb{R}^2$ exists satisfying*

$$\frac{\partial \gamma}{\partial t} = \kappa \cdot \mathbf{n}$$

where κ is the curvature of γ and \mathbf{n} is its unit inward normal vector. The curve γ is smooth for all t , it converges to a point as $t \rightarrow T$, and its limiting shapes as $t \rightarrow T$ is a round circle with convergence in the C^∞ norm.

References

- [1] DrexelNow, “Drexel’s image-tracking technology allows scientists to observe nature vs. nurture in neural stem cells,” *Science & Technology - Health*, Sep. 2015. [Online]. Available: <https://drexel.edu/now/archive/2015/September/CloneView/>.
- [2] V. Ulman, M. Maška, K. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, I. Smal, K. Rohr, J. Jaldén, H. Blau, O. Dzyubachyk, B. Lelieveldt, P. Xiao, Y. Li, S.-Y. Cho, and C. Ortiz-de-Solorzano, “An objective comparison of cell tracking algorithms,” *Nature Methods*, vol. 14, pp. 1–11, Oct. 2017. DOI: [10.1038/nmeth.4473](https://doi.org/10.1038/nmeth.4473). [Online]. Available: https://www.researchgate.net/publication/320722620_An_Objective_Comparison_of_Cell_Tracking_Algorithms.
- [3] “Evaluation methodology,” *Cell Tracking Challenge*, Jan. 2019. [Online]. Available: www.celltrackingchallenge.net/evaluation-methodology/.
- [4] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989. DOI: [10.1002/cpa.3160420503](https://doi.org/10.1002/cpa.3160420503). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160420503>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160420503>.
- [5] American Association for the Advancement of Science, “The cells in your body,” *Science Net Links*, Jun. 2018. [Online]. Available: <http://scienccenetlinks.com/student-teacher-sheets/cells-your-body/>.
- [6] D. C. Central, “Normal brain vs alzheimer’s,” *Learn About Dementia*, Aug. 2014. [Online]. Available: <https://www.dementiacarecentral.com/video/video-brain-changes/>.
- [7] M. Gage and R. S. Hamilton, “The heat equation shrinking convex plane curves,” *Journal of Differential Geometry*, vol. 23, no. 1, pp. 69–96, 1986. DOI: [10.4310/jdg/1214439902](https://doi.org/10.4310/jdg/1214439902). [Online]. Available: <https://doi.org/10.4310/jdg/1214439902>.
- [8] M. A. Grayson, “The heat equation shrinks embedded plane curves to round points,” *Journal of Differential Geometry*, vol. 26, no. 2, pp. 285–314, 1987. DOI: [10.4310/jdg/1214441371](https://doi.org/10.4310/jdg/1214441371). [Online]. Available: <https://doi.org/10.4310/jdg/1214441371>.
- [9] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*. Prentice-Hall, Inc, 1961, pp. 1–27.
- [10] A.-R. Mansouri, “Region tracking via level set pdes without motion computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 947–961, Aug. 2002. DOI: [10.1109/TPAMI.2002.1017621](https://doi.org/10.1109/TPAMI.2002.1017621). [Online]. Available: https://www.researchgate.net/publication/3193423_Region_tracking_via_level_set_PDEs_without_motion_computation.
- [11] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988, ISSN: 1573-1405. DOI: [10.1007/BF00133570](https://doi.org/10.1007/BF00133570). [Online]. Available: <https://doi.org/10.1007/BF00133570>.
- [12] E. K. Chong and S. H. Źak, *An Introduction to Optimization*, 4th. John Wiley & Sons, Inc, Feb. 2013, ISBN: 978-1-118-27901-4.
- [13] M. Kot, *A First Course in the Calculus of Variations*. American Mathematical Society, Oct. 2014, pp. 28–29, ISBN: 978-1-47041-495-5.
- [14] M. M. Denn, *Optimization by Variational Methods*. 1969, ISBN: 0882755951.
- [15] R. Kimmel, *Numerical Geometry of Images, Theory, Algorithms, and Applications*. Springer-Verlag New York, 2004, ISBN: 978-1-4684-9535-5. DOI: [10.1007/978-0-387-21637-9](https://doi.org/10.1007/978-0-387-21637-9).

- [16] S. C. Zhu and A. Yuille, “Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, Sep. 1996, ISSN: 19393539. DOI: [10.1109/34.537343](https://doi.org/10.1109/34.537343).
- [17] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd. Cambridge, U.K: Cambridge University Press, 1999, ISBN: 0521642043.
- [18] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. Springer New York, 1995, ISBN: 0387983465. DOI: [10.1007/978-1-4899-7278-1](https://doi.org/10.1007/978-1-4899-7278-1). [Online]. Available: <http://dx.doi.org/10.1007/978-1-4899-7278-1>.
- [19] M. E. Gage, “Curve shortening makes convex curves circular,” *Inventiones mathematicae*, vol. 76, no. 2, pp. 357–364, 1984, ISSN: 1432-1297. DOI: [10.1007/BF01388602](https://doi.org/10.1007/BF01388602). [Online]. Available: <https://doi.org/10.1007/BF01388602>.
- [20] R. Courant, H. Lewy, and K. Friedrichs, “On the partial difference equations of mathematical physics,” *New York University*, Apr. 1956. [Online]. Available: <https://archive.org/details/onpartialdiffere00cour/mode/2up>.
- [21] E. Bártová, G. Šustáčková, L. Stixová, S. Kozubek, S. Legartová, and V. Foltánková, “Recruitment of oct4 protein to uv-damaged chromatin in embryonic stem cells,” *PLOS ONE*, vol. 6, no. 12, pp. 1–13, Dec. 2011. DOI: [10.1371/journal.pone.0027281](https://doi.org/10.1371/journal.pone.0027281). [Online]. Available: <https://doi.org/10.1371/journal.pone.0027281>.
- [22] M. Cicconet, M. Gutwein, K. Gunsalus, and D. Geiger, “Label free cell-tracking and division detection based on 2d time-lapse images for lineage analysis of early embryo development,” *Computers in Biology and Medicine*, vol. 51, pp. 24–34, Aug. 2014, ISSN: 0010-4825. DOI: [10.1016/j.combiomed.2014.04.011](https://doi.org/10.1016/j.combiomed.2014.04.011). [Online]. Available: http://celltracking.bio.nyu.edu/?fbclid=IwAR0nTH_ILf7fIST9tOrBnDkw0_WBYrV9uQ8VnBr_Dtsyi_i0pit5_CjtCE4.
- [23] N. Allum, A. Allansdottir, G. Gaskell, J. Hampel, J. Jackson, A. Moldovan, S. Priest, S. Stares, and P. Stoneman, “Religion and the public ethics of stem-cell research: Attitudes in europe, canada and the united states,” English, *PLoS One*, vol. 12, no. 4, May 2017. [Online]. Available: <https://search.proquest.com/docview/1990009058?accountid=6180>.
- [24] B. von Tigerstrom, T. M. Nguyen, and B. M. Knoppers, “Regulation of stem cell-based therapies in canada: Current issues and concerns,” *Stem Cell Reviews and Reports*, vol. 8, no. 3, pp. 623–628, Mar. 2012, ISSN: 1558-6804. DOI: [10.1007/s12015-012-9360-0](https://doi.org/10.1007/s12015-012-9360-0). [Online]. Available: <https://doi.org/10.1007/s12015-012-9360-0>.
- [25] S. Li, B. C. Oh, C. Chu, A. Arnold, A. Jablonska, G. J. Furtmüller, H.-M. Qin, J. Boltze, T. Magnus, P. Ludewig, M. Janowski, G. Brandacher, and P. Walczak, “Induction of immunological tolerance to myelinogenic glial-restricted progenitor allografts,” *Brain*, vol. 142, no. 11, pp. 3456–3472, Sep. 2019, ISSN: 0006-8950. DOI: [10.1093/brain/awz275](https://doi.org/10.1093/brain/awz275). eprint: <https://academic.oup.com/brain/article-pdf/142/11/3456/30329908/awz275.pdf>. [Online]. Available: <https://doi.org/10.1093/brain/awz275>.
- [26] N. Yin, X. Yao, Z. Qin, Y.-L. Wang, and F. Faiola, “Assessment of bisphenol a (bpa) neurotoxicity in vitro with mouse embryonic stem cells,” *Journal of Environmental Sciences*, vol. 36, pp. 181–187, Oct. 2015, ISSN: 1001-0742. DOI: <https://doi.org/10.1016/j.jes.2015.06.004>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1001074215002776>.

- [27] L. Belkhir and A. Elmeligi, “Carbon footprint of the global pharmaceutical industry and relative impact of its major players,” *Journal of Cleaner Production*, vol. 214, pp. 185–194, 2019, ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2018.11.204>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0959652618336084>.
- [28] *The problem of pharmaceutical pollution - eeb - the european environmental bureau.* [Online]. Available: <https://eeb.org/the-problem-of-pharmaceutical-pollution/>.
- [29] *Summary of the pollution prevention act*, Aug. 2019. [Online]. Available: <https://www.epa.gov/laws-regulations/summary-pollution-prevention-act>.
- [30] P.-L. Cheah, L. M. Looi, and S. Horton, “Cost analysis of operating an anatomic pathology laboratory in a middle-income country,” *American Journal of Clinical Pathology*, vol. 149, no. 1, pp. 1–7, Dec. 2017, ISSN: 0002-9173. DOI: <10.1093/ajcp/aqx088>. eprint: <https://academic.oup.com/ajcp/article-pdf/149/1/1/23647934/aqx088.pdf>. [Online]. Available: <https://doi.org/10.1093/ajcp/aqx088>.
- [31] C. R. MacCluer, *Calculus of Variations: Mechanics, Control, and Other Applications*. Upper Saddle River, N.J: Pearson Prentice Hall, 2005, ISBN: 0-13-142383-5.
- [32] E. Saff and A. Snider, *Fundamentals of Complex Analysis with Applications to Engineering, Science, and Mathematics*, 3rd. Upper Saddle River, N.J: Prentice Hall, 2003, pp. 157–159, ISBN: 0139078746.
- [33] A. McLaren, “Ethical and social considerations of stem cell research,” *Nature*, vol. 414, no. 6859, pp. 129–31, Dec. 2001, ISSN: 1476-4687. DOI: <10.1038/35102194>. [Online]. Available: <https://doi.org/10.1038/35102194>.
- [34] M. R. Winter, M. Liu, D. Monteleone, J. Melunis, U. Hershberg, S. K. Goderie, S. Temple, and A. R. Cohen, “Computational image analysis reveals intrinsic multigenerational differences between anterior and posterior cerebral cortex neural progenitor cells,” *Stem Cell Reports*, vol. 5, no. 4, pp. 609–620, Oct. 2015, ISSN: 2213-6711. DOI: <10.1016/j.stemcr.2015.08.002>. [Online]. Available: <https://doi.org/10.1016/j.stemcr.2015.08.002>.
- [35] D. J. Struik, *Lectures on Classical Differential Geometry*, 2nd, ser. Dover books on mathematics. New York, NY: Dover, 1988, ISBN: 0-486-65609-8. [Online]. Available: <http://cds.cern.ch/record/2019240>.
- [36] T. Colding and W. Minicozzi, “Level set method for motion by mean curvature,” *Notices of the American Mathematical Society*, vol. 63, pp. 1148–1153, Nov. 2016. DOI: <10.1090/noti1439>.
- [37] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*. Springer-Verlag New York, Jan. 2006, ISBN: 978-0-387-44588-5. DOI: <10.1007/978-0-387-44588-5>.
- [38] A. Pressley, *Elementary Differential Geometry*, 2nd. Springer-Verlag London, 2010, ISBN: 978-1-84882-891-9. DOI: <10.1007/978-1-84882-891-9>.