Variational Methods and its Applications to Computer Vision

Erika Pellegrino

Imperial College London

A thesis submitted on time for the Degree of PhD

Abstract

Many computer vision applications such as image segmentation can be formulated in a "variational" way as energy minimization problems. Unfortunately, the computational task of minimizing these energies is usually difficult as it generally involves non convex functions in a space with thousands of dimensions and often the associated combinatorial problems are NP-hard to solve. Furthermore, they are ill-posed inverse problems and therefore are extremely sensitive to perturbations (e.g. noise). For this reason in order to compute a physically reliable approximation from given noisy data, it is necessary to incorporate into the mathematical model appropriate regularizations that require complex computations.

The main aim of this work is to describe variational segmentation methods that are particularly effective for curvilinear structures. Due to their complex geometry, classical regularization techniques cannot be adopted because they lead to the loss of most of low contrasted details. In contrast, the proposed method not only better preserves curvilinear structures, but also reconnects some parts that may have been disconnected by noise. Moreover, it can be easily extensible to graphs and successfully applied to different types of data such as medical imagery (i.e. vessels, hearth coronaries etc), material samples (i.e. concrete) and satellite signals (i.e. streets, rivers etc.). In particular, we will show results and performances about an implementation targeting new generation of High Performance Computing (HPC) architectures where different types of coprocessors cooperate. The involved dataset consists of approximately 200 images of cracks, captured in three different tunnels by a robotic machine designed for the European ROBO-SPECT project.

Declaration

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except where specified in the text. This dissertation is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other university.

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Erika Pellegrino
July, 2022

Contents

1 Introduction **17**

2 Mathematical Framework **27**

2.1 Variational Problems 28

2.1.1 Classical and Direct Methods in the Calculus of Variations 28

2.1.2 Free Discontinuity Problems 30

2.1.2.1 A General Introduction 30

2.1.2.2 The Space of Special Functions of Bounded Variation 35

2.1.2.3 Variational Convergence 37

2.1.2.4 The Mumford and Shah Functional 44

2.1.2.5 The Blake and Zisserman Functional 49

2.2 Variational Methods on Graphs 71

2.2.1 Discrete calculus 71

2.2.2 Regularization methods 72

2.2.2.1 Total variation 73

2.2.2.2 Max Flow - Min Cut 74

2.2.2.3 Combinatorial Dirichlet Problem 74

2.2.2.4 Free Discontinuity Problems on Graphs 75

2.2.3 Combinatorial Optimization Techniques 78

2.2.3.1 Maximum Flow/Minimum Cut Algorithms 78

2.2.3.2 Shortest Path Methods 79

2.2.3.3 Maximum Spanning Trees and Forests 80

2.3 Conclusions 80

3 Segmentation on Graphs **83**

3.1 Introduction 83

3.2 Classical Methods 85
3.2.1 Minimal Spanning Tree based methods 87
3.2.2 Graph Cut with Cost Functions 89
3.2.2.1 Minimal Cut Methods 89
3.2.2.2 Normalized Cut Methods 89
3.2.3 Graph Cut on Markov Random Field Models 92
3.2.3.1 Bi-labelling Graph Cut Methods 93
3.2.3.2 Multi-labelling Graph Cut Methods 94
3.2.3.3 Graph Cut with Shape Prior 95
3.2.3.4 Interactive Graph Cut Methods 95
3.2.4 Shortest Path Based Methods 96
3.2.5 Other Methods 98
3.3 Variational Methods 99
3.3.1 Merriman Bence Osher Scheme 101
3.3.1.1 Background 102
3.3.1.2 Algorithms 110
3.3.2 Total Variation Scheme 120
3.3.3 Mumford-Shah Scheme 122
3.4 Conclusions 128

4 Variational Restoration of Curvilinear Structures 129
4.1 Introduction 129
4.2 Curvilinear Structure Detection 130
4.2.1 Linear Filters 130
4.2.2 Non Linear Filters 134
4.2.3 Vesselness 137
4.2.4 Inclusion of vesselness in the variational model 139
4.3 Conclusions 147

5 Computer Vision Techniques for Inspection of Large Concrete Structures 149
5.1 Introduction 149
5.1.1 Autonomous Robotic Systems for Defect Detection 151
5.1.1.1 San Giorgio's Bridge 151
5.1.1.2 ROBO-SPECT Project 154

5.2 Crack Detection 158
5.2.1 Introduction 158
5.2.2 Automatic Crack Detection with Calculus of Variations . . 160
5.2.2.1 Evaluation Criteria 165
5.3 Digital Twin Model and 3D Point Cloud Segmentation 167
5.3.1 Datasets 173
5.3.2 Methods 174
5.4 Conclusions 180

6 Future Work **181**

Bibliography **183**

Glossary

$F^\star$ conjugate by duality. 139

$\Gamma^0(\mathbb{R}^n)$ lower semicontinuous convex functions. 138

ι_P indicator function of P. 139

$\mathcal{H}^{n-1}$ Hausdorff Measure. 34

∇ igradient operator. 44

BCDA block-coordinate descent algorithm. 40

CRF Conditional Random Field. 177

HPC High Performance Architectures. 3

MAP maximum a posterior. 90

MRF Markov Random Field. 90

MST minimal spanning tree. 85

Ncut normalized cut. 87

PCG preconditioned conjugate gradient. 58

SBV Special Functions of Bounded Variation. 30

List of Figures

1.1 Classification errors 18

1.2 Crack on a concrete wall ROBO-SPECT project 19

1.3 Neurites. (a) thinness and tortuosity. (b) overlapping 19

1.4 a image of the eye fundus showing the blood vessels 20

1.5 An image including several thin objects in a scanning electron microscopy image 21

1.6 Even careful thresholding does not allow both fibres to be included without noise coming from the background 22

2.1 Limitation of the Mumford-Shah model of detecting second-order geometrical features. (a,b) Gray-scale image with second-order edges. (c) Edge-detection via Mumford-Shah functional compared to (d) a full theoretical exact detection of 2nd-order features. . . . 52

4.1 The four adjacency graphs for the path opening algorithm. Graph (a) shows the S-N adjacency graph, graph (b) shows the SW-NE graph, (c) is the W-E adjacency, and (d) is the NW-SE adjacency graph 136

4.2 Ellipsoids representing the image structure. 145

5.1 Inspection with drones 150

5.2 RoboWash and RobotInspection 152

5.3 Inspection Bot: Full Extension of the Retractable Beam 153

5.4 The components of the ROBO-SPECT robotic platform 156

5.5 Left: Visual inspection of a tunnel at the Egnatia Motorway by the robotic paltform. Crack detection is driven by the presented computer vision component. Right: The robotic arm positioning the ultra-sonic sensor on a crack detected by the computer vision component 157
5.6 Illustration of a noisy case resulting in falsely identified cracked areas 157
5.7 Illustration of wrong crack position indication. The CNN annotations have falsely marked as potentially cracked areas noisy regions of the image. 158
5.8 Crack detection and properties retrieval results: (a) original crack image; (b) crack map; and (c) crack segments 159
5.9 Crack property retrieval procedure and results: (a) edge map; (b) crack skeleton; and (c) distance map 160
5.10 Crack on a concrete wall 164
5.11 Microcracks on a BSE microscopy of concrete materials 164
5.12 Mckanes Falls bridge 169
5.13 UAV photogrammetry vs Terrestrial Laser Scanning 170

List of Tables

5.1 Run time comparisons for a single image 163
5.2 Classification of the errors according to the value of the result and the ground-truth 165
5.3 Validation results of the proposed computer vision method using different types of metrics 166

Chapter 1

Introduction

Infrastructures can be exposed to different loading conditions, recurrent ones due to vehicular traffic and extraordinary ones caused by earthquakes, wind and strong rain. The consequently induced stresses may determine structural deterioration and damage, which can even cause catastrophic collapses [255]. Several authors have pointed out that increasing the level of automation for inspection and maintenance can play a crucial role in avoiding disasters, decreasing costs and increasing speed, accuracy and safety [174]. Recent works address the problem of monitoring tasks performed by robotic systems [199]. Existing ground or aerial solutions have been proposed for inspection of dangerous sites or those difficult to access, but at the present state-of-the-art, human-based procedures are not yet completely substituted. Examples of ground systems are wheeled robots [163], legged robots [119]. In case of inspection of vertical surfaces, wall-climbing robots were developed using magnetic devices [157] or using vacuum suction techniques [283]. Recently, unmanned aerial vehicles (UAVs) have shown a great potential in inspection applications due to their ability of collecting high-quality photo and video data [159].

Nowadays damages in buildings and bridges can be easily captured using a commercial digital camera and consequently analyzed by image processing algorithms. In this work data were provided by the European project ROBO-SPECT, whose objective is to design and implement an automated, faster and reliable tunnel inspection robotic system that detects cracks and other defects of the tunnel lining. The detection of cracks on a given Red Green Blue (RGB) image can be seen as an image segmentation problem: each pixel in the image is classified into

two classes; cracks and non-cracks. In the ROBO-SPECT machine vision module this classification process is conducted by a Convolutional Neural Network (CNN). However, the network annotation does not provide an accurate crack detection [260]. This is due to the fact that a deep learning classifier decides whether a pixel belongs to a crack region or not, based on the pixel intensity values of a restricted area centred around the candidate pixel. Tunnel images, however, are deeply affected by noise. These structures suffer from low lighting conditions, severe watering and from artificial structures that resembles crack lines. As a result, it is quite probable for the algorithm to confuse some of the candidate pixels from being cracks or not, especially if they belong to those "difficult" regions (see figure 1.1). In order to eliminate these "noisy areas" the aim of this work is to propose a variational method particularly effective in segmenting cracks in images (see section 2.1.2.5 and chapter 4).

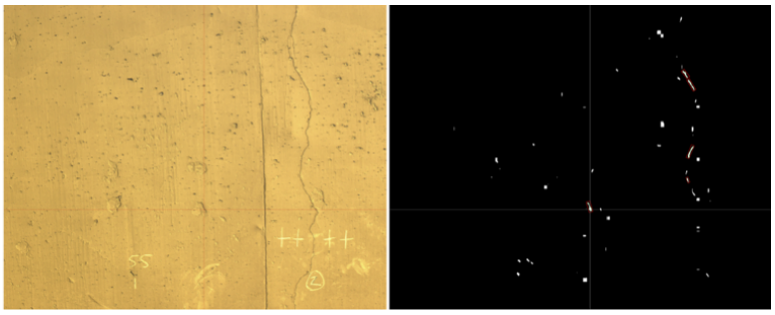


Figure 1.1: Classification errors

Cracks are particular type of curvilinear or thin structures. These types of objects play a main role in several fields as, for example, civil engineering, medical data analysis and computer assisted surgery:

- **Cracks in concrete**: Cracking is obtained by fast drying. Such cracks are sometimes smooth as they follow the contours of inclusions and sometimes very irregular, showing tunnels and disconnections. As a result, segmentation of such structures is particularly challenging as a result (see figure 1.2).

- **Neurites**: Neurites are a generic term to describe projections from cell bodies of neurons, i.e. axons and dendrites. Dendrites and axons in 2D microscopy tend to be quite tortuous. The individual neurites can generally



Figure 1.2: Crack on a concrete wall ROBO-SPECT project

assume a tree structure, although this is not strict. In addition, neurites seldom appear in isolation, leading to overlapping structures. In fluorescence modalities, disconnections are also common (see figure 1.3).

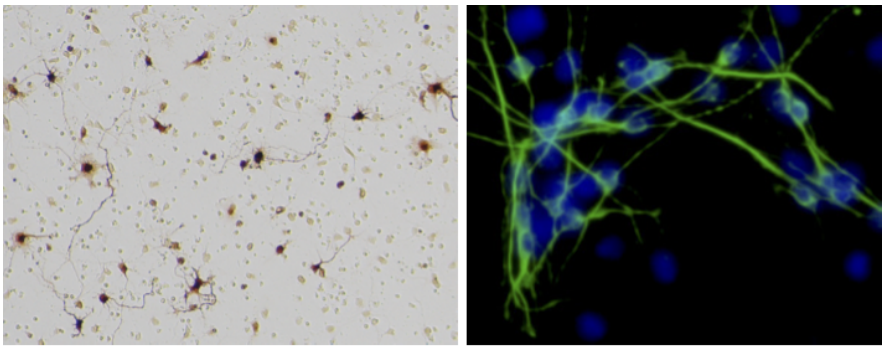


Figure 1.3: Neurites. (a) thinness and tortuosity. (b) overlapping

- **Retina blood vessels**: Images of the retina are useful for the diagnosis of many diseases. These include obviously eye-related conditions, such as macular degeneration, but also others that have circulatory implications such as diabetes. One reason for this is that the eye fundus shows the capillary network very well. Such images exhibit a very thin tree-like vascular structure over an uneven background. Segmenting this network of blood vessels is an important problem, particularly when looking for micro-aneurism, which are blockages in this network (see figure 1.4).

It does not exist an unambiguous definition of what constitutes a thin object,

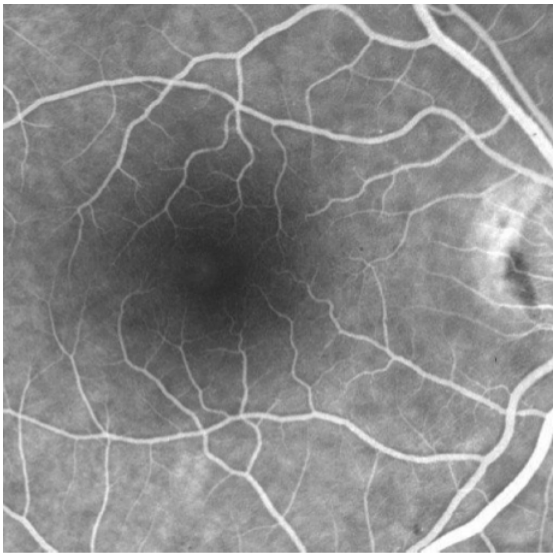


Figure 1.4: a image of the eye fundus showing the blood vessels

but we will say that an object is thin if it is semantically coherent and at least has one dimension smaller than the others. In other words, a thin object can be well-approximated by a submanifold of $\mathbb{R}^n$, where n is 2 or 3 typically, of lower dimension than n. Contrariwise, we will say that an object is "isotropic" if it is not-thin, i.e. all of its dimensions are comparable (see 1.5). Many image analysis applications involve the segmentation of these structures: cracks in material science images, vessels in medical images and roads in remote sensing images. In the general literature on image processing and analysis, thin objects are not usually mentioned as deserving special treatment. However, because they possess this dimension that is much smaller than the others, they are indeed often harder to acquire, process, segment and analyse than more "isotropic" objects. For image filtering, for instance, one often uses masks or windows of some fixed dimension. Usually, one makes the assumption that these masks entirely fits into most of the objects of interest. This may not be true for thin object. In segmentation, many popular methods assume that one can start from some starting point in an object and propagate information around until object contours are found. For thin objects, one may not be able to perform this propagation and moreover, one may not even be able to define their contours. Moreover, a single object in an image is assumed to be connected, but due to noise and discretization issues, often thin objects might be locally disconnected. Whereas a human observer

might still recognize semantically a crack, a fibre or a vessel, at the local pixel level, the information is often lost in the noise. How then is it possible to connect high-level information to low-level vision operators ? This is a prevalent problem in computer vision and image analysis, but it is particularly difficult to solve for thin objects. These structures require specific segmentation procedures because they are prone to extra problems when compared with more "compact objects". First of all, they are very sparse, which means that the ratio of pixels containing information is very small. This obviously complicates the task, in particular for statistical methods and learning approaches. Moreover, this sparsity makes the evaluation of segmentation algorithms very difficult. This is a frequently underestimated, but crucial, consequence. Indeed, without good performance measures, comparing and improving curvilinear structure segmentation is not straightforward.

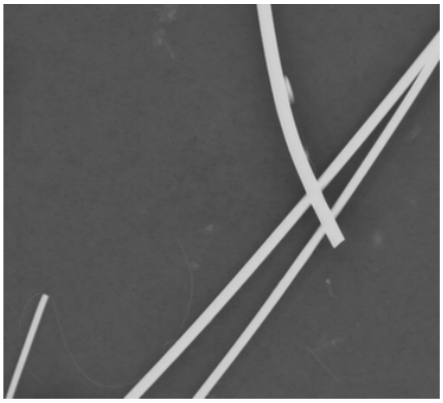


Figure 1.5: An image including several thin objects in a scanning electron microscopy image

Secondly, due to their low thickness, these structures are also more sensitive to noise and artifacts (see figure **??**). Even a small amount of noise may be sufficient to disrupt their contours, leading to disconnections which make the segmentation task even more challenging. Once again, classical quality scores do not measure the preservation connectivity. Finally, curvilinear structures in real applications generally present complex geometries and topology. They can be more or less tortuous, present different orientations and scales inside the same image, and also form a network (possibly with cycles), making geometric priors very difficult to use.



Figure 1.6: Even careful thresholding does not allow both fibres to be included without noise coming from the background

The chapter **Mathematical Framework** is mainly concerned with variational methods. In computer vision the variational approach provides a way to return an approximation of an image which is a composition of several pieces on which the image values are *homogeneous* in term of derivatives. Two well-known variational models due to Mumford and Shah (MS) [229] and Blake and Zisserman (BZ) [35] are investigated because they are promising towards thin structures. Segmentation can be intuitively defined as the process of partitioning a domain into disjoint and homogeneous regions while detecting the regions boundaries. From the mathematical and engineering point of view a signal, either in one or in two dimensions, can be considered as a function $g(x)$, where x is a point of the domain Ω and the value $g(x)$ is the signal value at the point x. In image analysis, segmentation is useful as a pre-classification process producing a smooth image, with preserved regions boundaries, which is somehow easier to classify than the original image. More generally, in signal analysis the process of segmentation does allow to identify the main signal features by smoothing the input data $g(x)$, i.e. reducing its noise level and preserving, i.e. not smoothing, the data discontinuities so that the signal meaningful structure is preserved and more easily observable. Back to mathematics, the segmentation process can be formalised, among others, in a *variational framework*. Within the framework of the *Calculus of Variations* it is possible to state the segmentation concept as a minimum problem, that is to find a solution minimising a defined quantity that is a summation of penalty terms associated to the required solution features. The variational nature of the Mumford and Shah model and the Blake and Zisserman

model can be basically understood considering that the penalty terms involved in both models take care of different requirements. In fact, the segmentation criteria require: *a)* the solution to be *as close as possible* to the input data; *b)* the solution to be *as smooth as possible* within each homogeneous region, that is the solution has to be less noisy than the original data and the smoothing must not be imposed on the regions boundaries since they contain relevant information that must not be lost; *c)* the length of the regions boundaries to be *as short as possible*, that is the regions boundaries have to be as smooth as possible as well. The concept $as \cdot aspossible$ is strictly related to the choice of approach the segmentation problem in a variational framework, i.e. as a minimum problem, where the concepts of *close*, *smooth* and *short* have to find a proper mathematical formalisation. In particular, the Mumford and Shah model is defined by a functional where three penalty terms are considered: the first term penalises the *distance* between the approximating solution and the data, the second term penalises the *wiggle* of the approximating solution and the third term penalises the *length* of the region boundaries. The Blake and Zisserman model is defined by a functional where a term still penalises the *distance* between the approximating solution and the data, a term still penalises the *length* of the region boundaries and the smoothness of the solution is controlled by an higher order term with respect to the one in Mumford and Shah model. Moreover, the *wiggle* of the regions boundaries is also penalised by an ad hoc term. Each penalty term is associated with a weight, represented by a real parameter, controlling the relative influence of the penalty terms and their overall effects on the solution. It is important to notice that the dimension of the term controlling the regions' boundaries is one order less than the order of the penalty term controlling the distance between the approximating solution and the data and of the penalty term controlling the smoothness of the approximating solution. Moreover, since both the Mumford and Shah functional and the Blake and Zisserman functional segment the data into homogeneous regions and detect the region boundaries explicitly and directly, the solution is made of two elements: one is properly the approximating solution, u and the other is the set K of closed curves representing the region boundaries. This kind of problems, where the unknown is a pair (u, K), with K varying in a class of closed subset of a fixed open set $\Omega \subset \mathbb{R}^n$, and $u : \Omega \setminus K \to \mathbb{R}^n$ is a function in some function space, are generally denoted by "Free Discontinuity Problems". Usually they are of the form:

$$min\{E_v(u,K) + E_s(u,K)\},$$

with $E_v(u,K), E_s(u,K)$ being interpreted as *volume* and *surface* energies, where, in any case $E_v(u,K)$ is defined in a dimension one order greater that the dimension of $E_s(u,K)$. For instance, the Mumford and Shah model requires the minimisation of a functional of the form:

$$\mathcal{MS}(u,K) = \int_\Omega |u-g|^2\,dx + \int_{\Omega\setminus K} |\nabla u|^2\,dx + \mathcal{H}^{n-1}(K\cap\Omega) \qquad (1.1)$$

The first term penalises the distance between the approximating solution u and the data g. The second term penalises the gradient of the approximating solution strictly within the homogeneous regions, that is the solution is required to be smooth within each homogeneous region and it is allowed to present strong transitions along the regions' boundaries. The third term penalises the length of the regions' boundaries being the length measured by the $(n-1)-$Hausdorff measure of the set K. The main result of this chapter is a parallel algorithm for these minimization problems. This is of remarkable importance when huge amount of data is considered.

The chapter **Segmentation on Graphs** shows several segmentation techniques formulated on graphs with emphasis on variational models. Designing formulations of variational methods on graphs [152] has numerous advantages over schemes like finite difference or finite elements. Firstly, it removes the necessity to discretize the data again in a different form. Secondly, it is possible to reuse existing efficient combinatorial optimization tools to minimize energies (i.e. network flows, shortest paths, minimum spanning trees) and develop new ones. Finally, the generality of these approaches makes it possible to extend the developed techniques from images to a larger class of data (i.e. mesh filtering and data classification). In Section 3.3.3 we proposed a graph model based on the mathematical framework showed in Chapter 2 for denoising point cloud data. Given a point cloud in a Euclidean space with (noisy) real-valued labels or an undirected graph with labeled vertices, the model denoises the labels while allowing for jumps (discontinuities) in label values.

The chapter **Variational restoration of curvilinear structures** describes how to build a variational framework that would be particularly effective towards

the segmentation of thin structures. Upon the notion of mathematical morphology it is possible to define two curvilinear characteristics: an intensity feature, that can be seen as a curvilinearity measure and a directional feature, that provides a local orientation estimation of the curvilinear structures. Both features are low-level structure characteristics and are essential devoted to be embedded in more sophisticated image processing methods for segmentation of thin structures. In Paragraph 4.2.2, we proposed a graph version of the traditional path opening algorithm.

The chapter **Computer Vision Techniques for Inspection of Large Concrete Structures** shows some useful applications to defect detection in large concrete structures and point cloud clustering. In particular, a parallel numerical procedure for crack detection is described. Performances on data provided by a robotic machine have been collected on several HPC architectures aiming at reducing the execution time in a significant way. These methods are described in the following publications [247],[248],[249], [246], [250], [251].

Chapter 2

Mathematical Framework

This Chapter presents a review of the main concepts and tools that are used to study the Mumford and Shah and the Blake and Zisserman problems both from the theoretical point of view, i.e. to prove that the functional involved admit a minimum, and also from the practical point of view, i.e. to build a convenient numerical approximation of these functionals. In particular, the main purpose of this chapter is to analyse the proposed mathematical framework and to show the connection between the originally stated problems and the approximation that are eventually computed.

The *classical* and the *direct* method in the **Calculus of Variations** are briefly reviewed as they are the main tools to explicitly compute the solution and to study the existence of minima. The existence of the solution is guaranteed by a generalisation of the Weierstrass Theorem, however, there are problems for which the existence conditions are not satisfied and for which the direct methods cannot be applied. In these cases a so-called *relaxation* is needed to associate to the original problem a weak formulation, which is analytically tractable: Free Discontinuity Problems belong to this category. For this reason is necessary to develop mathematical tools that allow the definition of *approximating* formulations that are numerically more tractable. It is important to notice that many different types of *approximations* can be defined and that, for all of them, the asymptotical equivalence to the original problem is guaranteed and proved by the use of the Γ-convergence theory.

2.1 Variational Problems

The problem of finding, among all functions with prescribed boundary conditions, those which minimise a given integral functional is one of the main goal in the **Calculus of Variations**. In general, the problem can be formalised as follows:

$$\min\{\mathcal{F}(u) : u \in X\} \tag{2.1}$$

where X is a suitable Banach space and

$$\mathcal{F}(u) = \int_{\Omega} f(x, u(x), \nabla u(x))dx \tag{2.2}$$

where $\Omega \subset \mathbb{R}^n$ is a bounded open set with boundary $\partial\Omega, u : \Omega \subset \mathbb{R}^n \to \mathbb{R}, \nabla u \in \mathbb{R}^n, f : \Omega \times \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ is a continuous function.

2.1.1 Classical and Direct Methods in the Calculus of Variations

In general, *classical* methods allow to explicitly compute the solution of a minimum problem like 2.1 defined above by means of the derivation of the Euler equation associated to $\mathcal{F}$. *Direct* methods allow to formally prove the existence of the minimum and are generalisation of the Weierstrass Theorem. This guarantees the existence of the solution under the conditions that the functional $\mathcal{F}$ is lower semicontinuous and that the function space X is compact. Regarding the *classical* methods in the **Calculus of Variations**, the solution of the minimisation problem can be obtained by finding the zeros of the "derivative" of $\mathcal{F}, \mathcal{F}'(u) = 0$ known as the *Euler equation* associated to the functional $\mathcal{F}$. If $u \in C^2$ the Euler equations are:

$$\mathcal{F}'(u) = -\sum_{i=1}^{n} \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial \xi_{ij}}(x, u, \nabla u)\right) + \frac{\partial f}{\partial u_j}(x, u, \nabla u)$$

where $f = f(x, u, \xi)$. As examples, the Euler equations associated to the problem of minimising the Dirichlet integral and to the minimal surface problem are here

reported. The Dirichlet problem is

$$\min\left\{\mathcal{F}(u)=\int_\Omega |\nabla u|^2 dx : u=u_0 \text{ on } \partial\Omega\right\}$$

where $n \geq 1$ and the associated Euler equation is:

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \\ u = u_0 & \text{on } \partial\Omega \end{cases}$$

The minimum surface problem is:

$$\min\left\{\mathcal{F}(u)=\int_\Omega \sqrt{1+|\nabla u|^2} dx : u=u_0 \text{ on } \partial\Omega\right\}$$

where $n \geq 1$ and the associated Euler equation is

$$\begin{cases} -\sum_{i=1}^n \frac{\partial}{\partial x_i}\left[(1+|\nabla u|^2)^{-1/2}\frac{\partial u}{\partial x_i}\right] = 0 & \text{in } \Omega \\ u = u_0 & \text{on } \partial\Omega \end{cases}$$

A very strong link exists between **Partial Differential Equations** and **Calculus of Variations**. In fact, variational problems produce, via their associated Euler equation, differential equations and, on the other side, many differential equations can be studied by variational methods. For a treatment on this topic, specifically oriented to image processing problems we refer to [19] and to [7] where a wider class of problems is presented. The *direct* methods in **Calculus of Variations** deal directly with the functional $\mathcal{F}$ to prove the existence of a minimum. The existence can be proved by defining a minimising sequence from which, under specific conditions on the function space, it is possible to extract a convergent subsequence which, under some other conditions on the continuity of $\mathcal{F}$, converges to a minimum of the functional $\mathcal{F}$. In practice, the functional $\mathcal{F}$ has to be lower semicontinuous with respect to the topology defined on the function space X, the characteristics of the topology have to ensure the compactness of the minimising sequences. In terms of functions, let X be a metric space, and let $\overline{R} = R \cup \{-\infty, \infty\}$, a function $F : X \to \overline{R}$ is said to be lower semicontinuous if:

$$\mathcal{F}(x) \leq \lim_{h\to\infty} \inf \mathcal{F}(x_h)$$

for every sequence (x_h) converging to x and for all $x \in X$. A subset K of X is compact if every sequence in K has a subsequence which converges to a point of K i.e.

$$\forall\,(x_h) \subset K \quad \exists x \in X, \exists\,(x_{h_k}) : x_{h_k} \to x$$

A function $F : X \to \overline{R}$ is coercive if the closure of $\{F \leq t\}$ is compact in X for every $t \in \mathbb{R}$.

The *Weierstrass theorem* guarantees that if F is coercive and lower semicontinuous then:

i. F has a minimum point in X;

ii. If x_h is a minimising sequence of F in X, and x is the limit of a subsequence of x_h then x is a minimum point of F in X;

iii. if F is not identically $+\infty$, then every minimising sequence for F has a converging subsequence.

The *direct* methods of the **Calculus of Variations** cannot always be applied directly to approach a minimum problem. This may occur when the functional $\mathcal{F}$ is not lower semicontinuous or the space X is not compact. When $\mathcal{F}$ is coercive, but not lower semicontinuous a widely adopted strategy to solve the minimum problem is to associate to the functional $\mathcal{F}$ another one, called $\mathcal{R}_F$ (relaxed functional). This new functional defines a new problem, (R_P), called relaxed problem. In particular $\mathcal{R}_F$ admits a minimum and has the following two properties:

- $\min\{\mathcal{R}_F\} = \inf\{\mathcal{F}\}$;
- the minimum points for $\mathcal{R}_F$ are the limits of minimising sequences of $\mathcal{F}$, and every minimising sequence of $\mathcal{F}$ has a subsequence converging to a minimum point of $\mathcal{R}_F$.

The relaxed functional $\mathcal{R}_F$ is defined as the greatest lower semicontinuous functional less or equal to $\mathcal{F}$.

2.1.2 Free Discontinuity Problems

2.1.2.1 A General Introduction

The terminology "Free Discontinuity Problems" has been introduced by De Giorgi to indicate a class of variational problems characterised by a competition between

volume energies, concentrated on a n-dimensional set, and surface energies, concentrated on a $(n-1)$-dimensional set. A relevant feature of these problems is that the support K of the surface energies is not fixed a priori and it is in many cases a relevant unknown of the problem. Free discontinuity problems involve functionals whose natural domains are sets of functions which admit a finite number of discontinuities, being K the set of the discontinuities. Since the discontinuity set K is not necessarily made of closed curves, i.e. boundaries, the class of free discontinuity problems presents only some analogies with the class of the free boundary problems and requires a specific mathematical theory. Some of the main examples where free discontinuity problems arise are:

- fracture mechanics;
- theory of plasticity;
- optimal partitions;
- signal and image reconstruction;

In particular, the segmentation model proposed by Mumford and Shah is one of the best known free discontinuity problems. Approaching a free discontinuity problem, and in particular the Mumford and Shah problem, following the *direct* methods in the **Calculus of Variations** presents many difficulties mainly related to the dependence of the involved energies on the surface K. The definition of a relaxed problem associated to the original free discontinuity problem is hence necessary to allow the proof of the existence of the solution. Again, even when the existence of the solution is guaranteed, the exact computation of the solution can be rarely performed. This motivates the need to approximate the relaxed problem with a functional which solution can be computed practically. The effectiveness of the approximation is formally proved through the use of the Γ-convergence techniques. The value of the solution can be computed, for example, applying the *classical* methods in the **Calculus of Variations**, i.e. deriving the Euler equation associated to the approximating functional. Moreover, the approximation is in general numerically tractable, i.e. solution can be computed automatically. The approximation of free discontinuity problems can be achieved following many different techniques. Some of the more relevant are:

- non-local approximation;

- finite-difference approximation;
- finite-elements approximation;
- slicing method;
- second order singular pertubation approximation;
- elliptic approximation

For a review of these methods we refer to [48], [9] and [19]. Approximation techniques include both discrete methods, (such as the finite elements approximation) and methods in the realm of functional problems that must be further approximated by a discretization to obtain a numerical solution. In the following some notions about the space of Special Functions of Bounded Variation (SBV) and about the Γ-convergence theory are briefly sketched. The SBV space provides an unified framework for the study of free discontinuity problems and it allows a weak formulation of otherwise intractable problems. The Γ-convergence theory allows to prove the approximation of a given functional by a new functional, defined on different function spaces, which is more tractable from both the analytical and the numerical viewpoint.

Regularization, Weak Derivatives and Discontinuity Sets Visual reconstruction leads to inverse mathematical problems which are generally ill-posed. Regularization provides a method to make such problems well-posed. The basic idea of regularization methods is to restrict the class of admissible solutions of an ill-posed problem by imposing additional constraints. In the standard Tikhonov theory, a problem is made well-posed by restricting the solutions to a space of smooth functions. Through regularization, an ill-posed problem may be reformulated as a problem of calculus of variations. A numerical solution can be computed using finite difference or finite elements methods. However, the smoothness constraint is inadequate in the presence of visual discontinuities that are the most significant locations in any image since they often indicate the boundaries of objects. For this reason is necessary to extend the standard regularization methods in order to take discontinuities into account. Mumford and Shah and also Blake and Zissermann proposed a variational approach to image segmentation.

Standard regularization methods require the minimization of an elliptic functional. Variational problems for regularization with discontinuities involve both an analogous elliptic functional and a measure of the length of the discontinuity contour. The discontinuity curves are themselves among the unknows of the problem and this makes the minimization of the functional difficult. The concept of variational convergence provides a solution for this problem. It proposes to approximate a variational problem by a sequence of different problems. A particular concept of variational convergence for functionals, the Γ-convegence [90] is suitable for problems of calculus of variations with free discontinuities. The classical theorem proved by Ambrosio Tortorelli shows how a functional depending on discontinuities can be approximated by a sequence of elliptic functionals which are more tractable. The Γ-convegence makes it possible to go to the limit in the corresponding minimization problems: the minimizers of the functionals of the sequence converge in a appropriate metric to the minimizer of the original functional.

In the standard Tikhonov regularization theory, the class of admissible solutions of an ill-posed problem is restricted to Sobolev space of smooth functions. Regularization with discontinuities requires a more general space of the Bounded Variation functions (BV). A function of two variable $u(x, y)$ is called a function of bounded variation in the domain Ω if it is summable and there exists a constant K such that for any h_1 and h_2 the following inequality hold:

$$\iint_{\Omega_h} |u(x + h_1, y + h_2) - u(x, y)|\, dx\, dy \leq K||h||$$

where $||h||$ denotes the norm of the vector h with components (h_1, h_2) and Ω_h is the subset of points of Ω whose distance from the boundary of Ω is greater than the norm of h. The functions of bounded variation form a Banach vectorial space. Mumford and Shah [229] proposed a variational formulation for the problem of image segmentation, which is referred to as the "weak membrane" model by Blake and Zisserman. Their approach can be generalized to other reconstruction problems. A variational method for the regularization of ill-posed vision problems involving discontinuities looks for a BV function $u(x, y)$ which minimizes the

following functional:

$$E(u) = \iint_{\Omega} \Phi(u, x, y)\, dx\, dy + \lambda \iint_{\Omega} ||\nabla u||^2\, dx\, dy + \alpha \mathcal{H}^1(S_u)$$

where S_u is the set of discontinuities points of $u(x, y)$, $\mathcal{H}^1(S_u)$ is the one-dimensional Hausdorff measure of S_u, constants λ and α are positive weights and $\Phi(u, x, y)$ is a non-negative function. If S_u is the union of rectifiable curves, the Hausdorff measure is simply the total length of the set of curves. If $\Phi(u, x, y) = (u - f)^2$, with $f(x, y)$ representing the data, the corresponding functional is called the weak membrane functional by Blake and Zisserman. The first term measures the discrepancy between the solution $u(x, y)$ and the input data. The second term obliges the solution to be smooth outside the discontinuity set S_u. By minimizing the total length of the curves along which the solution is discontinuous the third term prevents the formation of an incoherent discontinuity set. The contour length has been chosen because it is the simplest reasonable measure of the set of jump points. It should be noted that the gradient of a BV function has a generalized meaning with respect to the one used in mathematical analysis. It can be shown that the derivative (in the distributional sense) of a BV function can be decomposed into a regular part, whose density is a summable function called the approximate differential, and a singular part. Only the approximate differential appears in the second integral in the expression of $E(u)$. The weights λ and α are the parameters of the problem. Blake and Zisserman found that the weight λ is a scale parameter, while the square root of the ratio $2\alpha/\lambda^{1/2}$ is a threshold which determines the detection of an isolated step edge. The edge is detected if the step height exceeds such a threshold. The parameter α is a measure of the resistance to noise. The presence of noise in the data will generate the appearance of spurious discontinuities in the solution. As α increases (for a given λ) the probability of spurious discontinuities decreases. The value of α should be proportional to the variance of the noise. Blake and Zisserman also found that the error in the localization of the discontinuities is negligible if the signal-to-noise ratio is not too small. The parameter λ plays an important role in the areas where the solution is smooth, preventing random disturbances occasioned by the presence of noise. The value of this parameter should be set in inverse proportion to the signal-to-noise ratio in the input data. The problem of minimizing functional $E(u)$ can be considered as an extension of the classical

regularization method in the presence of discontinuities. It has not been proved if this problem is well-posed, but it can be conjectured that this is true. Ambrosio [5] proved that functionals of this type are lower semicontinuous, and hence have minimizers, in a suitable subclass of the functions of bounded variations, denoted by SBV (special functions of bounded variation). The SBV functions have the property that the singular part of the distributional derivative is concentrated along the discontinuty set (like a δ distribution). Unfortunately, the SBV space also includes functions with highly irregular discontinuity sets, which may contain non rectifiable curves (this fact justifies the Hausdorf measure in the definition of the functional). The computation of the discontinuity contours makes the minimization of the functional $E(u)$ a difficult problem, both theoretically and practically, because the set S_u of the discontinuity curves is an unknown of the problem. The concept of Γ -convergence proved by Ambrosio and Tortorelli, allowing the approximation of functional $E(u)$ by elliptic functionals, makes the problem numerically more tractable.

2.1.2.2 The Space of Special Functions of Bounded Variation

One of the innovative contribution by De Giorgi to the study of the free discontinuity problems indicates to interpret K as the set of discontinuity points of the function u and to define u on a particular space of discontinuous functions. The definition of such a function space fulfils the following requirements:

- it has to be possible to define K as a smooth set of discontinuity points of the function u;
- u has to be "differentiable" almost everywhere outside K, so that the bulk energy depending on ∇u can be defined;
- the possibility to apply the direct methods in the Calculus of Variations has to be guaranteed;

The rigorous definition and the formalisation of such a space is due to the efforts of De Giorgi and Ambrosio who defined the space of **Special Functions of Bounded Variation** (SBV). Working in this space has proved to be particularly fruitful allowing the definition of the so called "weak forms" of original problems and providing the necessary tools to study the regularity of the solutions of relaxed

problems. Various regularity results shown that a weak formulation, defined in the SBV space, does provide a solution to a wide class of free discontinuity problems. The SBV space is the function space that is commonly used in image analysis, the main reason is that, as opposed to classical Sobolev spaces, functions in SBV can be discontinuous across hypersurfaces. As for images this means that images are discontinuous across edges. We report here a rough definition of the space of Functions of Bounded Variation (BV) and of the space of Special Functions of (SBV), we refer to [9] and [48] for the formal definitions and details on such spaces.

Let $\Omega \subseteq \mathbb{R}^n$ be an open set, let $u : \Omega \to \mathbb{R}$ be a measurable function and $x \in \Omega$. Let $u^+(x)$ and $u^-(x)$ denote, respectively, the approximate upper and the lower limit of u at x, defined by:

$$u^+(x) = \inf \left\{ t \in \mathbb{R} : \lim_{\rho \to o^+} \frac{|\{y \in \Omega : |x-y| < \rho, u(y) > t\}|}{\rho^n} = 0 \right\}$$
$$u^-(x) = \sup \left\{ t \in \mathbb{R} : \lim_{\rho \to o^+} \frac{|\{y \in \Omega : |x-y| < \rho, u(y) < t\}|}{\rho^n} = 0 \right\}$$

If $u^+(x) = u^-(x) \in \mathbb{R}$, then x is said to be a *Lebesgue point* of u and the common value of $u^+(x)$ and $u^-(x)$ is called *approximate limit* of u at x. By definition u is a function of bounded variation in Ω, if $u \in L^1(\Omega)$ and its *distributional derivative* is a vector-valued measure Du with total variation $|Du(\Omega)|$ defined by:

$$|Du(\Omega)| = \sup \left\{ \int_\Omega u \; div\Phi : \Phi \in C_0^1(\Omega, \mathbb{R}^n), |\Phi| \leq 1 \right\}$$

The space of all functions of bounded variation on Ω will be denoted by $\mathrm{BV}(\Omega)$. If $u \in \mathrm{BV}(\Omega)$, then S_u is countable $\mathcal{H}^{n-1}, n-1$ rectifiable, i.e.

$$S_u = N \cup \bigcup_{i \in \mathbb{N}} K_i$$

where $\mathcal{H}^{n-1}(N) = 0$, and each K_i is a compact set contained in a C^i hypersurface.

The distributional derivative Du of a function $u \in \mathrm{BV}(\Omega)$ can be decomposed as $Du = D^a u + D^s u$ where D^a is absolutely continuous and $D^s u$ is singular with respect to the Lebesgue measure. The density of $D^a u$ with respect to the Lebesgue measure is denoted by ∇u. Moreover, $D^j u$ denotes the restriction of $D^s u$ to S_u and $D^c u$ the restriction of $D^s u$ to $\Omega \setminus S_u$. With these notations the

distributional derivative of u can be decomposed as

$$D_u = D^a u + D^j u + D^c u$$

where $D^j u$ and $D^c u$ are respectively called the *jump* part of Du and the *Cantor* part of Du. Here it is important to note that the measure $D^j u$ is concentrated on S_u. A function u belongs to SBV when the Cantor part $D^c u$ of its distributional derivatives D_u is null[1]. A generalisation of SBV spaces, namely the space of Generalised Special functions of Bounded Variation (GSBV), is necessary when dealing with functions $u \notin L^1(\Omega)$. The space GSBV is defined

$$GSBV = \{u : \Omega \to \mathbb{R} : \text{ Borel function, } -k \vee u \wedge k \in SBV_{loc}(\Omega) \forall k \in \mathbb{N}\}$$

where $SBV_{loc}(\Omega)$ denotes the class of functions $v \in \text{SBV}(\Omega^{'})$ for every $\Omega^{'} \subset\subset \Omega$. This is the case, for example, of the space where a solution of the Blake and Zisserman problem has to be searched. In particular, the existence of minimisers of the Blake and Zisserman functional has been proved by [58] in the space $GSBV^2 \cap L^2(\Omega)$ where

$$GSBV^2(\Omega) = \{u : \Omega \to \mathbb{R} : u \in GSBV(\Omega), \nabla u \in [GSBV]^n\}$$

2.1.2.3 Variational Convergence

A new concept of convergence for sequences of functionals has appeared in mathematical analysis in recent years. This concept is specially designed to approach the limit in the sequences of the corresponding variational problems and is called variational convergence. This notion of convergence can be used to approximate one variational problem by another, which may have quite different computational properties. When applied to minimization problems, the convergence of the sequence of functionals $F^k(u)$ to the limit functional $F(u)$, in a variational sense, requires that minimizers of $F^k(u)$ converge (in a suitable metric) to minimizers of $F(u)$ as k tends to infinity. Furthermore, the minimal values of the $F^k(u)$ must converge to the minimal value of $F(u)$. Hence, a variational convergence is a weak notion of convergence for sequences of functionals which makes

[1] A function $u \in \text{SBV}(\Omega)$ can be represented as $u^a + u^j \in W^{1,1}(\Omega)$ and $u^j \in X(\Omega)$ the space of SBV function whose derivative reduces to the jump part, i.e. in $X(\Omega)$, $\nabla u = 0$ and $D^j(u)$ is purely atomic measure

it possible to go to the limit in the corresponding minimization problems.

A particular theory of variational convergence, the Γ-convergence introduced by De Giorgi [90], has an interesting computational application to problems of calculus of variations with free discontinuity sets. A sequence of functionals $F^k(u)$ defined on a metric space U is said to be Γ-convergent to the functional $F(u)$ if the following two conditions hold for all $u_0 \in U$:

i. for every sequence u_k converging to u_0 (in the metric of the space U) one has:

$$\liminf_{k\to\infty} F^k(u_k) \geq F(u_0) \tag{2.3}$$

ii. there exists a sequence u_k converging to u_0 such that:

$$\limsup_{k\to\infty} F^k(u_k) \leq F(u_0) \tag{2.4}$$

The limit functional $F(u)$ is called the $\Gamma - limit$ of the sequence $F^k(u)$. The Γ-limit when it exists is unique. The meaning of the inequality 2.3 is the following: if the limit of the numerical sequence $F^k(u)$ exists, it is greater than or equal to $F(u_0)$; otherwise, the limit of every convergent subsequence $F^{k_i}(u_{k_i})$ is greater than or equal to $F(u_0)$. The inequality 2.4 likewise means that all limit values of the numerical sequence $F^k(u)_k$ are less than or equal to $F(u_0)$. It should be noted that $\Gamma - convergence$ is a different concept from pointwise convergence. More in depth, Γ-convergence is not implied by and does not imply pointwise convergence. They are two independent concepts, comparable in the sense that if $F(u)$ and $F_1(u)$ are, respectively, the Γ-limit and the pointwise limit of sequence $F^k(u)$, then $F(u) \leq F_1(u)$ for every $u \in U$.

The fundamental variational property of Γ-convergence can now be formulated: let $F^k(u)$ be a sequence of functionals defined on the metric space U which is Γ-convergent to the limit functional $F(u)$ as k tends to infinity. Assume also that the functionals $F^k(u)$ have minimizers in U. It can then be shown that if a sequence u_k^* of minimizers of $F^k(u)$ converges, then the limit is a minimizer of $F(u)$, and $F^k(u_k^*)$ converges to the minimal value of $F(u)$. Hence, the Γ-convegence is a notion of variational convergence. If k is sufficiently large, the problem of minimizing $F(u)$ can then be replaced by the problem of minimizing $F^k(u)$.

There is another property of Γ-convergence which is of relevance to the calculus of variations. If $F^k(u)$ Γ-converges to $F(u)$, it can then be proven that $F^k(u) + G(u)$ Γ-converges to $F(u) + G(u)$ for every continuous functional $G(u)$. This property means that Γ-convergence is stable under continuous perturbations. This stability feature plays an important role in the application of the theorem of Ambrosio and Tortorelli to the problem of the minimization of the functional $E(u)$.

In recent years, variational principles with a free discontinuity set have been introduced to solve reconstruction problems in computer vision theory (see, for instance [224, 132, 265]). The variational approach to the image segmentation problem proposed by Mumford and Shah [229] consists of minimizing the functional

$$E(u, K) = \int_{\Omega \setminus K} \left(|\nabla u|^2 + \mu |u - g|^2 \right) dx + \alpha \mathcal{H}^{n-1}(K \cap \Omega)$$

where $\Omega \subset \mathbb{R}^n$ is a bounded open set, $\mathcal{H}^{n-1}$ is the Hausdorff $(n-1)-$dimensional measure, $g \in L^\infty$, and $\alpha, \mu > 0$ are fixed positive parameters. The functional has to be minimized over all closed sets $K \subset \overline{\Omega}$ and all $u \in C^1(\Omega \setminus K)$. In the case $n = 2$ the function g represents the image to be segmented. By minimizing the functional one tries to detect the discontinuities of g due to noise and small irregularities. The set K contains the jump points of u and represents the edges of the objects. The functional penalizes large sets K, and outside K the function u is required to be close to g and C^1. The Mumford and Shah variational principle can be extended to several reconstruction problems of computer vision: stereo reconstruction [289], computation of optical flow [231], shape from shading [290]. Variational problems involving functionals of this form are usually free discontinuity problems, after terminology introduced by De Giorgi [89]. The Mumford and Shah model has some drawbacks: it is unable to reconstruct crease discontinuities and yields the over-segmentation of steep gradients (the so-called ramp effect). To overcome these defects of the first order model. Blake and Zisserman [35] introduced a second order functional which can be written in the form:

$$\begin{aligned} F(u, K_0, K_1) = & \int_{\Omega \setminus (K_0 \cup K_1)} \left(\left| \nabla^2 u \right|^2 + \Phi(x, u) \right) dx \\ & + \alpha \mathcal{H}^{n-1}(K_0 \cap \Omega) + \beta \mathcal{H}^{n-1}((K_1 \setminus K_0) \cap \Omega) \end{aligned}$$

with $\alpha, \beta > 0$ positive parameters. The functional has to be minimized over the

unknown sets K_0, K_1 with $K_0 \cup K_1$ closed and $u \in C^2(\Omega \backslash (K_0 \cup K_1))$ approximately continuous on $\Omega \backslash K_0$. If $\Phi(x,u) = \mu|u-g|^2$ and $n = 2$ the functional 2.1.2.3 is just that one introduced in [35] (the thin plate surface under tension). In the second order model, K_0 represents the set of jump points for u and $K_1 \setminus K_0$ is the set of crease points, which is particularly relevant in those computer vision problems that require the reconstruction of thin structures. If the conditions (see[35])

$$\beta \leq \alpha \leq 2\beta$$

are satisfied, the existence of minimizers for the functional $F(u, K_0, K_1)$ has been proven, in the case $n = 2$ and $\Phi(x,u) = \mu|u-g|^2$, by Carriero, Leaci and Tomarelli [59] (notice that 2.1.2.3 are necessary and sufficient for the lower semicontinuity of $\overline{F}$ with respect to the L^1 convergence). The proof is based on a weak formulation of the problem by setting

$$\bar{F}(u) = \int_\Omega \left(\left| \nabla^2 u \right|^2 + \Phi(x,u) \right) dx + \alpha \mathcal{H}^{n-1}(S_u) + \beta \mathcal{H}^{n-1}(S_{\nabla u} \backslash S_u),$$

where ∇u denotes an approximate differential, S_u is the discontinuity set of u in an approximate sense, and $S_{\nabla u}$ is the discontinuity set of ∇u. In [58] the existence on minimizers for the functional $\overline{F}$ over the space

$$\left\{ u : \Omega \to \mathbf{R} : u \in L^2(\Omega), u \in GSBV(\Omega), \nabla u \in [GSBV(\Omega)]^n \right\}$$

has been proven in any space dimension n, $GSBV(\Omega)$ being the space of generalized special functions of bounded variation introduced in [94]. A regularity theorem in [59] then shows that, for $n = 2$, any weak minimizer actually provides a minimizing triplet (u, K_0, k_1) of F by taking a suitable representative of the function and the closure of S_u and S_∇. Ambrosio and Tortorelli [10] approximated the Mumford and Shah functional by a family of elliptic functionals defined on Sobolev spaces. The approximation takes place in a variational sense, the De Giorgi Γ-convergence. The approximating elliptic functionals proposed in [11] are defined by

$$E_\epsilon(u,s) = \int_\Omega (s^2 + \lambda_\epsilon) |\nabla u|^2 \, dx + \mu \int_\Omega |u-g|^2 \, dx + \alpha G_\epsilon(s)$$

where the approximation takes place as $\epsilon \to 0^+, \lambda_\epsilon \to 0^+$, and

$$G_\epsilon(s) = \int_\Omega \left[\epsilon |\nabla s|^2 + \frac{(s-1)^2}{4\epsilon} \right] dx$$

The variable $s \in [0,1]$ is related to the set of jumps K. The minimizing s_ϵ are near to 0 in a neighbourhood of the set K, and far from the neighbourhood they are close to 1. The neighbourhood shrinks as $\epsilon \to 0$. The Ambrosio and Tortorelli's approximation can be used to find an effective algorithm for computing the minimizers of E 2.1.2.3. The approximation has been applied to several computer vision problems.

We now present a similar technique for the Blake and Zisserman functional. We consider the following family of functionals

$$F_\epsilon(u,s,\sigma) = \int_\Omega (\sigma^2 + \kappa_\epsilon) |\nabla^2 u|^2 \, dx + \int_\Omega \Phi(x,u) \, dx + (\alpha - \beta) G_\epsilon(s) \\ + \beta G_\epsilon(\sigma) + \xi_\epsilon \int_\Omega (s^2 + \zeta_\epsilon) |\nabla u|^\gamma \, dx$$

for suitable infinitesimals $\kappa_\epsilon, \xi_\epsilon, \zeta_\epsilon$. A slight variant of these functionals has been proposed by Bellettini and Coscia [26] in the case $n = 1$ and in that case the Γ-convergence of F_ϵ to $\overline{F}$ has been proved. Their result can be extended in the following way: the lower inequality of Γ-convergence is proven in any space dimension n, and the upper inequality is proven when u is bounded and $|\nabla u| \in L^\gamma(\Omega)$, under a very mild regularity assumption on the sets S_u and $S_\nabla u$, which is fulfilled in computer visions applications. In the particular case when $\alpha = \beta$ and $n = 2$ the full Γ-convergence is proven.

The extension of the Ambrosio and Tortorelli's approximation to the second order problem presents several difficulties. The lower inequality cannot be obtained by means of the slicing technique and consequent reduction to a one-dimensional problem used in [26]. Such a reduction yields the operator norm of the Hessian matrix in the Γ-limit instead of the Euclidean norm. The second derivatives are then estimated by adapting a global technique proposed by Ambrosio in [6] and relying on a compactness theorem in the space 2.1.2.5 due to Carriero, Leaci and Tomarelli [58]. Conversely, the jump part of the functional is estimated by using a slicing argument, taking into account that the space $GSBV$ is a vector space under a suitable energy condition.

The major difficulty in the proof of the upper inequality consists in obtaining a suitable estimate on $\int |\nabla u|^\gamma \, dx$ from the finiteness of 2.1.2.3. Such an estimate would permit to adapt the constructive part of Ambrosio and Tortorelli's proof to the second order problem. In the case $\alpha = \beta, n = \gamma, \gamma = 2$, an estimate which yields a full Γ-convergence result is obtained by means of a suitable interpolation inequality in the Sobolev space $W^{2,2}$. If $\alpha \neq \beta$ the upper inequality can be proved under the assumption that u is bounded and $|\nabla u| \in L^\gamma(\Omega)$.

The discretization of the functional 2.1.2.3 is not straightforward and it is difficult to apply gradient descent with respect to the unknown sets K_0 and K_1. Conversely, the Γ-convergent approximation yields a sequence of functionals 2.1.2.3 which are numerically more tractable, so that discretization and gradient descent may be applied in a straightforward way. In particular, a recent numerical minimization technique of non-convex functional based on an especially tailored version of a block-coordinate descent algorithm (BCDA) can be applied. Although theoretical models such as Mumford and Shah and as Blake and Zisserman are global, the non-convexity of the objective functionals forces numerical methods to provide sub-optimal solutions. The outcome of many numerical experiments has highlighted that, although the theoretical model is global, the solutions weakly depend on boundary conditions and they are energetically close to initial data. This fact motivates the development of a tiling scheme in order to address the segmentation of large images where a minimizer of the functional is assembled by merging together local minimizers restricted to sub-portions of the image. The notion of Γ-convergence is of particular interest in solving minimum problems since it can be used to describe the asymptotic behaviour of this class of problems and it formalises a notion of variational convergence. The theory of Γ-convergence allows to prove the equivalence between two minimum problems, one characterised by a functional $\mathcal{F}$ and another characterised by a a functional $\mathcal{F}_\epsilon$. Some problems are in fact originally formulated through a functional $\mathcal{F}$ and the Γ-convergence allows its approximation by means of a sequence of a more tractable functional $\mathcal{F}_\epsilon$. The free discontinuity problems are problems of this kind. On the other hand, there are other problems originally formulated by a functional $\mathcal{F}_\epsilon$ that is difficult to handle numerically when the parameter ϵ is small. In this case the Γ-convergence is used to prove that the original functional can approximated by a parameter-free functional. The homogenization problems are classical examples of this kind. The usefulness of proving such an equivalence

lies on the fact that, depending on particular cases under consideration, one of the two functionals is more tractable that the other, both from an analytical and a numerical viewpoint. It is worth to provide a brief list of some of the more relevant problems that can be faced and solved by using Γ-convergence technique. Examples includes:

- the gradient theory of phase transition;
- homogenization problems;
- dimension reduction,
- continuous limits of differences schemes;
- approximation of Free Discontinuity Problems;

We report in the following just the definition of Γ-convergence and a relevant result related to the Γ-convergence, while for an introductory treatment on the theory of Γ-convergence and for a parade of examples we refer to [49]. Let (X, d) be a metric space and let $f, f_\epsilon : X \to [0, \infty]$ be functions. The sequence of functions (f_ϵ) Γ-converges to f if the following two conditions are satisfied:

(i) for any sequence $(x_\epsilon) \subset X$ converging to x the following holds:

$$\liminf_{\epsilon \to 0} f_\epsilon (x_\epsilon) \geq f(x)$$

(ii) for any $x \in X$ there exists a sequence $(x_\epsilon) \subset X$ converging to x such that

$$\limsup_{\epsilon \to 0} f_\epsilon (x_\epsilon) \leq f(x)$$

The function f is uniquely determined by (i), (ii) and is denoted by $\Gamma - \lim_{\epsilon \to 0} f_\epsilon$, moreover if $f(x_\epsilon)$ is a converging sequence to x so that

$$\lim_{\epsilon \to 0} f_\epsilon (x_\epsilon) = \lim_{\epsilon \to 0} f_\epsilon (x)$$

then its limit is a minimum point for f.

2.1.2.4 The Mumford and Shah Functional

One of the best known and studied free discontinuity problem is the model introduced by Mumford and Shah ([229]) for image segmentation. The model is presented here to review the main aspects related to the formulation of such a problem in a variational framework. Segmentation is a typical and widely investigated topic in computer vision. It can be defined as the process of partitioning an image into groups of pixels that represent objects. A class of segmentation methods recognizes the objects as the regions delimited by edge-boundaries, i.e., sets of pixels presenting sharp variations of intensity. Mathematical methods for segmentation are mainly divided into two categories: methods based on PDEs and variational methods. PDE's approaches originated in the early 80s with the isotropic scale-space noise-reduction coarsing based on heat diffusion proposed in [333]. Then, anisotropic diffusion has been introduced, which inhibits diffusion according to local properties of the image [253]. Another approach, which is a particular case of anisotropic diffusion, has the interesting property that the related PDE represents the flow generated by the minimization of the Total Variation [270]. Principal issues of PDEs approaches are mainly due to the difficult interpretation of the role that parameters play in the model and the physical meaning of solutions [325]. In this perspective, the variational approach seems to be more intuitive and allows for having a proper and explicit modelling of all the components: noise-reduction, edge-detection, scale-space representation. By fully exploiting a variational framework, Mumford and Shah [229] proposed a model for image segmentation based on the minimization of the following functional

$$\mathcal{MS}(u,K) = \int_{\Omega\setminus K} |\nabla u|^2\,dx + \alpha\mathcal{H}^1(K\cap\Omega) + \mu\int_{\Omega}|u-g|^2\,dx$$

Here $\Omega\subset\mathbb{R}^2$ and $g\in L^\infty(\Omega)$ is the input image. The minimization is among all the functions continuously differentiable outside K, i.e. $u\in C^1(\Omega\setminus K)$, where $K\in\overline{\Omega}$ is compact. $\mathcal{H}^1$ is the $1-$dimensional Hausdorff measure, and α,μ are positive parameters. The minimization of the first term forces u to be smooth (a piecewise constant behaviour is expected) outside K. Because of the term $\mathcal{K}\cap\Omega^1$, K is a one-dimensional set with finite length. The last integral term is a distance term that forces u to be close to the original image g. The set K can be easily understood as the set of the discontinuities of u, indeed this is a typical problem

belonging to a general class of problems called *free discontinuities problems* [89]. Kawohl found a strict relationship between the Mumford-Shah and Perona-Malik approaches to segmentation [178]. In particular, he showed how the parameters of the Mumford-Shah (MS) functional can be interpreted as parameters regulating an anisotropic diffusion process applied to the image g.

From a practical point of view, the minimization of the MS functional 2.1.2.4 cannot be addressed because the measure term $\mathcal{H}^1(K \cap \Omega)$ is not semi-continuous with respect to any reasonable topology. As suggested in [89], by relaxing the problem into the weaker space of Special Functions of Bouded Variation $SBV(\Omega)$, the methods of Calculus of Variations can be used to prove the existence of minima [88]. The advantage of this approach is that for every $u \in SBV(\Omega)$, the discontinuity set S_u is uniquely determined by geometrical properties of the functions. This results in a functional formulation of the MS problem that uniquely depends on the function u:

$$\mathcal{G}(u) = \int_\Omega |\nabla u|^2 \, dx + \alpha \mathcal{H}^1(S_u \cap \Omega) + \mu \int_\Omega |u - g|^2 \, dx$$

where $u \in SBV(\Omega)$ and S_u is the complement set of Lebesgue points of u. Using compactness and lower semi-continuity theorems [132] it is showed that under mild conditions, there exists a solution such that $\mathcal{H}^1(S_u) < \infty$. Moreover, by regularity results one has that $\mathcal{H}^1(\overline{S_u} \setminus S_u) = 0$ and the pair $(u, \overline{S_u})$ can be identified with a minimizer of the strong formulation. Based on this relaxed formulation, many techniques have been proposed to tackle the problem of numerically computing a minimizer. The free discontinuity term poses a serious problem. Ambrosio and Tortorelli [9], by exploiting a nice result of Modica and Mortola [220], proposed a Γ-convergence approximation via integral functionals defined on proper Sobolev spaces. In their approximation the discontinuity set is replaced by an auxiliary function that plays the role of indicator function. Numerical solutions based on the Ambrosio-Tortorelli approximation are given in the framework of Finite Element Method (FEM) in [25], and via finite-difference discretization of Euler-Lagrange equations in [104]. In [70], a Γ-convergence approximation using local integral functionals defined on a discrete space is given. Numerical implementation of the method is presented in [40]. Another minimization technique is based on a convex relaxation of the functional [256]. A level set approach to minimization is presented in [73]. With no intent of being exhaustive, we refer

the interested reader to the overview on the numerical approaches for solving the MS functional given in [48].

Numerical Implementations The Mumford-Shah functional [229] is among the most influential publications in the field of image processing. This and related publications by Blake and Zisserman [35] and others have sparked enormous research activity on discontinuity-preserving smoothing, piecewise-smooth approximations and minimal partition problems [224]. Yet, the computation of the *piecewise-smooth* approximation has rarely made it into practical image and video analysis methods because minimization of this non-convex functional is difficult and existing algorithmic solutions are far from real-time capability.

The Mumford-Shah functional provides a prototypical form of all regularizers which aim at combining a smoothing of homogeneous regions with the enhancement of edges. Given a bounded open set $\Omega \subset \mathbb{R}^d$,$d \geq 1$, the vectorial Mumford-Shah problem is given by

$$\min_{u,K} \left\{ \int_\Omega |u-f|^2 dx + \alpha \int_{\Omega \setminus K} |\nabla u|^2 dx + \lambda |K| \right\}$$

where $f : \Omega \to \mathbb{R}^k$ is a vector-valued input image with $k \geq 1$ channels. This model approximates f by a function $u : \Omega \to \mathbb{R}^k$ which is smooth everywhere in Ω except for a possible $(d-1)$-dimensional jump set K, at which u is discontinuous. The weight $\lambda \geq 0$ controls the length of the jump set K (less jumps for larger λ) and $\alpha \geq 0$ penalizes the smoothness of u outside of K. The limiting case $\alpha \to \infty$ imposes zero gradient ∇ outside K and is known as the *piecewise constant* Mumford-Shah model or "cartoon" limit. The norm of the gradient $|\nabla u|$ is the Euclidean norm $|\nabla u|^2 = \sum_i |\nabla u_i|^2$, and the norm in the term $|u-f|$ is also Euclidean.

The Mumford-Shah problem has been intensively studied in the applied mathematics community [224]. In practice its applicability is substantially limited because of its non-convexity. While it is often replaced by the convex total variation, this is a poor substitute because of its tendency to reduce the contrast at edges and oversmooth flat regions (staircasing). As a consequence, researchers have developed different optimization strategies to tackle the non-convex Mumford-Shah problem.

Alternating Minimization Schemes One kind of methods consists of

non-convex approximations of the original Mumford-Shah functional, where one alternating minimizes for u and for K [10, 73].

Ambrosio-Tortorelli Approach. The non-convex phase-field model of Ambrosio and Tortorelli [10], for example, is given by:

$$\min_{u,s} \int_\Omega |u-f|^2 dx + \alpha \int_\Omega (1-s)^2 |\nabla u|^2 dx + \lambda \int_\Omega \left(\varepsilon |\nabla s|^2 + \frac{1}{4\varepsilon} s^2 \right) dx \qquad (2.5)$$

with a small parameter $\epsilon \geq 0$. The key idea is to introduce the additional variable $s : \Omega \to \mathbb{R}$ as an edge set indicator, in the sense that points $x \in \Omega$ are part of the edge set K if $s(x) \approx 1$ and part of the smooth region if $s(x) \approx 0$. It was shown in [10] that this approximation Γ-converges to the Mumford-Shah functional for $\epsilon \to 0$, i.e. minimizers of 2.5 approach the minimizer of 2.1.2.4. One finds u and s by alternating minimization, computing s for fixed u and viceversa. Each of these subproblems is elliptic and can be solved quickly, e.g. by the linearly converging primal-dual method [71]. Extensions of this approximation to color images have been proposed in [51]. One disadvantage of this model, beside its non-convexity, is its dependency on an additional parameter ϵ. To obtain a good approximation of minimizers of 2.1.2.4, ϵ must be chosen small for increasing α, and for large α the dependency becomes sensitive and a good choice is unclear. This makes the approach unfeasible for the piecewise constant case $\alpha = \infty$.

L_0 Smoothing Approach of Xu et al. For the piecewise constant case, Xu et al. [337] recently proposed a fast approximating method. Assuming the image domain has been discretized into a finite rectangular grid, again denoted by Ω, the piecewise constant Mumford-Shah limit corresponds to L_0 penalization of the gradient:

$$\min_u \sum_{x \in \Omega} |u(x) - f(x)|^2 + R_{MS_0}(\nabla u(x)) \qquad (2.6)$$

where

$$R_{MS_0}(g) = \begin{cases} \lambda & \text{if } g \neq 0 \\ 0 & \text{else} \end{cases} \qquad (2.7)$$

and the gradient $\nabla u = (\nabla u_i)_{1 \leq i \leq k}$ is discretized e.g. using forward differences. Intuitively, the regularizer R_{MS_0} summed up over all pixels counts how many times u changes its value. This way, it prefers regions of constancy instead of smooth variations. Xu et al. propose a quadratic decoupling strategy to solve

2.6, introducing new variables g which approximate the gradient:

$$\min_{u,g} \sum_{x \in \Omega} |u(x) - f(x)|^2 + \beta |\nabla u(x) - g(x)|^2 + R_{MS_0}(g(x)) \tag{2.8}$$

with a parameter $\beta \geq 0$ and the Euclidean norm for the coupling. This approximation is again solved via alternating minimization. After having computed the next u and g, the parameter β is increased to $k\beta$ with a $k \geq 1$ until a final β_{max} is reached. Starting value for β is chosen automatically as $\beta_0 = 2\lambda$. Multiplier k is set either to 2 (fast but smooth result) or 1.05 (slow and more piecewise constant). Because of the empirical nature of the coupling, it is not clear how the computed solutions u mathematically relate to the original model 2.6, or even to 2.1.2.4. In fact, the computed solutions are not piecewise constant, but vary smoothly over large areas.

Convex Relaxation Methods In the recent past, several authors have overcome the issue of non-convexity by suggesting convex relaxations for respective functionals [138, 3]. Convex relaxations for the piecewise constant Mumford-Shah functional were proposed in [194, 69, 341]. Convex relaxations for the piecewise smooth Mumford-Shah model were proposed for the scalar and the vectorial case. The key idea is to rewrite the multi-label problem as a binary labeling problem in a higher-dimensional space. Relaxation of the binary constraint leads to a convex problem which can be minimized optimally. Subsequent binarization provides an approximate solution of the original problem. Some of these approaches were clearly inspired by the Markov Random Field (MRF) community, where the discrete variant of the Mumford-Shah regularizer is typically referred to as a truncated quadratic penalizer [46, 187, 315].

Unfortunately, these methods to compute approximate minimizers are currently far from real-time capability because the added label space dimension drastically increases memory and run time. For the Mumford-Shah model the run time even grows quadratically in the number of considered color values. Thus, the problem of computing good approximate minimizers of the Mumford-Shah energy in real-time remains an important challenge.

2.1.2.5 The Blake and Zisserman Functional

The Mumford and Shah model has some drawbacks: it is unable to reconstruct crease discontinuities and yields the over-segmentation of steep gradients (the so-called ramp effect). To overcome these defects of the first order model, Blake and Zisserman [35] introduced a second order functional which can be written in the form:

$$F(u, K_0, K_1) = \int_{\Omega \setminus (K_0 \cup K_1)} (|\nabla^2 u|^2 + \Phi(x, u))\, dx + \alpha \mathcal{H}^{n-1}(K_0 \cap \Omega) + \beta \mathcal{H}^{n-1}((K_1 \setminus K_0) \cap \Omega)$$

with $\alpha, \beta \geq 0$ positive parameters. The functional has to be minimized over the unknown sets K_0, K_1, with $K_0 \cup K_1$ closed and $u \in C^2(\Omega \setminus (K_0 \cup K_1))$ approximately continuous on $\Omega \setminus K_0$. If $\Phi(x, u) = \mu|u - g|^2$ and $n = 2$, the functional 2.1.2.5 is just that one introduced in [35] (the thin plate surface under tension). In the second order model, K_0 represents the set of jump points for u and $K_1 \setminus K_0$ is the set of crease points. Since the reconstruction of crease discontinuities is particularly relevant in computer vision problems which require the reconstruction of visible surfaces from two-dimensional images, the function Φ was introduced. A suitable choice of this function will allow to apply this variational method to problems as for instance the computation of the depth from pairs of stereo images. If the conditions $\beta \leq \alpha \leq 2\beta$ are satisfied, the existence of minimizers for the functional $F(u, K_0, K_1)$ has been proved, in the case $n = 2$ and $\Phi(x, u) = \mu|u - g|^2$ by Carriero, Leaci and Tomarelli [59]. The proof is based on a weak formulation of the problem by setting

$$\overline{F}(u) = \int_\Omega (|\nabla^2 u|^2 + \Phi(x, u))\, dx + \alpha \mathcal{H}^{n-1}(S_u) + \beta \mathcal{H}^{n-1}(S_{\nabla u} \setminus S_u)$$

where ∇u denotes an approximate differential, S_u is the discontinuity set of u in an approximate sense, and $S_{\nabla u}$ is the discontinuity set of ∇u. In [58] the existence of minimizers of the functional $\overline{F}$ over the space

$$\left\{ u : \Omega \to \mathbb{R} : u \in L^2(\Omega), u \in GSBV(\Omega), \nabla u \in [GSBV(\Omega)]^n \right\}$$

has been proved in any space dimension n, $GSBV(\Omega)$ being the space of generalized special functions of bounded variation introduced in [94]. A regularity theorem in [59] shows that, for $n = 2$, any weak minimizer actually provides a minimizing triplet (u, K_0, K_1) of F by taking a suitable representative of the function and the closure of S_u and $S_{\nabla u}$. The discretization of this functional is difficult, but it is possible due to the Γ-convergence.

Second Order Models and Thin Structures Being a first-order model, the MS variational segmentation suffers of some side effects [35]. The minimization of the gradient norm forces the solution to be locally constant (zero gradient) and to result in a step-wise function where the gradient of is too steep. This phenomenon is well-known as over-segmentation of steep gradients. Moreover, the minimization of the length term results in an approximation of complex edge junctions by triple-junctions where edges meet at $2/3$ wide angles. This may lead to a degradation of the real geometry of boundaries. Lastly, properly because of its first-order nature, the MS model is unable to detect second-order geometrical features such as points of gradient discontinuity, see 2.1. Since very often such points correspond to thin structures, the MS model has the limitation that is not capable of detecting them. With the specific intent to overcome such problems, Blake and Zisserman proposed a variational model based on second order derivatives, free discontinuities and free gradient discontinuities [35]. In their original formulation one has to minimize

$$BZ(u, K_0, K_1) = \int_{\Omega\setminus(K_0\cup K_1)} (|\nabla^2 u|^2\, dx + \mu \int_\Omega |u - g|^2\, dx + \alpha\mathcal{H}^1(K_0 \cap \Omega) + \beta\mathcal{H}^1((K_1 \setminus K_0) \cap \Omega)$$

among all functions u that are twice differentiable (with continuity) outside $K_0 \cup K_1$ and at least differentiable (with continuity) outside K_0. K_0 and K_1 vary among all the compact sets such that $K_0 \cup K_1$ is closed in $\overline{\Omega}$. μ, α, β are positive parameters. Here $|\nabla^2 u|^2$ denotes the Hessian matrix of u. Notice that, for an admissible solution u, discontinuities are allowed both on $K_0 \cup K_1$, whereas discontinuities of the gradient are allowed only on K_1 . α and β are contrast parameters regulating the total length of the discontinuity sets. From practical point of view, the minimization of the functional cannot be addressed because the measure term $\mathcal{H}^1(K_0 \cap \Omega)$ is not semi-continuous with respect to any reasonable

topology. As suggested in [89], by relaxing the problem in to the weaker space of Generalized Special functions of Bounded Variation $GSBV(\Omega)$, the methods of Calculus of Variations can be used to prove the existence of minima [88, 58]. In this space a relaxation of the functional 2.1.2.5 is given by:

$$\mathcal{F}(u) = \int_{\Omega} (|\nabla^2 u|^2 + \mu|u - g|^2)\, dx + (\alpha - \beta)\mathcal{H}^1(S_u) + \beta\mathcal{H}^1((S_{\nabla u} \cap S_u)$$

where $u \in GSBV^2(\Omega) = \{w \in GSBV(\Omega) : \nabla w \in [GSBV(\Omega)]^2\}$. In this weaker space, a proper definition of $\nabla^2 u$ and $S_{\nabla u}$ (the theoretic discontinuity set of ∇u) as geometrical property of the function u, is possible. By regularity arguments it can be proved [59] that a minimizer of 2.1.2.5 can be identified with a minimizing couple of the strong formulation, provided $\beta \leq \alpha \leq 2\beta$. Thus, the original optimal set $K_0 \cup K_1$ is recovered via the discontinuity set S_u and the gradient discontinuity set $S_{\nabla u}$.

A vivid research interest is devoted to the Blake-Zissermann functional as it represents the generalization of the well-known a widely used Mumford-Shah. From a theoretical point of view it is a challenging topic, since well-posedness of the problem and uniqueness of the solution [37] as well as regularity properties of minimizers [57, 60, 61] are still under investigation. Segmentation based on the Blake-Zisserman model, because of its second-order nature, is specifically suitable for addressing problems such as image inpainting [62], where the functional minimization allows for predicting partially occluded regions in an image and their contours continuation. One of the scopes of this work is to propose a novel variational method to detect thin structures such as cracks on surfaces. If singularities related to edges are classically associated with a discontinuity of gray level intensities across edges (and are thus detected using spatial gradient information carried by the image), this characterization proves to be unsuitable when dealing with points, cracks or filaments. Indeed, while for an edge the singularity is associated with a jump of the intensity across this edges, for filaments, such a jump does not occur. A heuristic illustration of this fact is given by an approximation of the 1D function defined by $f(x) = 0$ if $x \neq 0$ and $f(0) = 1$ as

follows:

$$f_\eta(x) = 0 \text{ if } |x| \geq \eta$$
$$f_\eta(x) = \frac{2}{\eta^3}|x|^3 - \frac{3}{\eta^2}|x|^2 + 1 \text{ if } |x| \leq \eta$$

It is not difficult to see that $f_\eta(x)' = 0$ showing that the differential operator of order 1 does not capture the singularity at 0. On the other hand, as $f_\eta(x)'' = -\frac{6}{\eta^2}$, f_η'' clearly exhibits a singularity at 0 when η becomes small. This exemplifies the fact that in order to detect fine structures or filaments, higher order differential operators should be considered. This intuitive illustration can be then formalized in 2D in the following way: assuming that a crack can be modelled by an indicator function supported by a smooth curve Γ, it can be approximated by a sequence of smooth functions whose Hessian matrices blow up in the perpendicular direction to Γ, while their gradient is null. Motivated by these observations showing that a suitable model should involve higher order derivatives, the crack recovery model we propose falls within second order variational models. It is based on the Blake-Zisserman functional for computer vision problems that depends on free discontinuities, free gradient discontinuities and second order derivatives, and more precisely, on its approximation by elliptic functionals defined on Sobolev spaces.

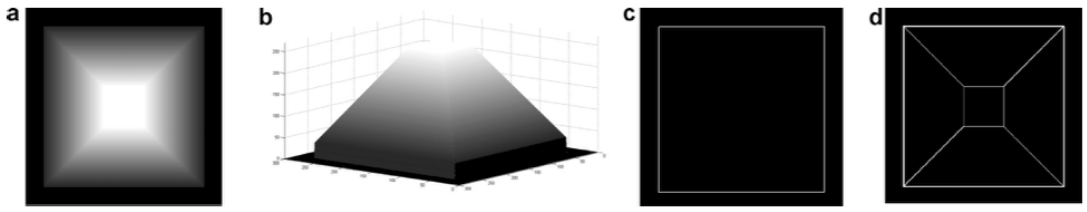


Figure 2.1: Limitation of the Mumford-Shah model of detecting second-order geometrical features. (a,b) Gray-scale image with second-order edges. (c) Edge-detection via Mumford-Shah functional compared to (d) a full theoretical exact detection of 2nd-order features.

Numerical Implementations Implementing gradient descent of 2.1.2.5 with respect to the unknown free discontinuity sets is extremely difficult. Γ-convergence has shown to be fundamental to solve the problem of numerically computing a minimizer. This notion of convergence, suitable for functionals, has been introduced by De Giorgi and Franzoni [90]. For a deep treatment of this topic we refer to [85, 49]. The key point in Γ-convergence is that a specific functional,

which may not have good properties for minimization, can be approximated by a sequence of regular functionals all admitting minimizers. The sequence of the approximate minimizers converges (in the classical sense) to a minimizer of the original objective functional. Besides its importance as mathematical tool, Γ-convergence is very attractive also from a numerical point of view as it allows for the solution of several difficult numerical problems in Computer Vision, Physics and many other fields. See for instance [49]. Following the idea of Ambrosio and Tortorelli, in [26] a Γ-convergence result is proved for the BZ functional in dimension 1. A full proof in dimension 2 and a partial result for any dimension n is given by Ambrosio [8]. The authors, by properly adapting the techniques of [26] and [10], have introduced two auxiliary functions $s, z : \Omega \to [0, 1]$ (aimed at approximating the indicator functions of the discontinuity sets) to the model and proposed a Γ-convergence approximation of $\mathcal{F}$ via the family of uniformly elliptic functionals

$$\mathcal{F}_\epsilon(s, z, u) = \delta \int_\Omega z^2 \left|\nabla^2 u\right|^2 dx + \xi_\epsilon \int_\Omega \left(s^2 + o_\epsilon\right) |\nabla u|^2 dx + (\alpha - \beta) \int_\Omega \epsilon |\nabla s|^2 + \frac{1}{4\epsilon}(s-1)^2 dx + \beta \int_\Omega \epsilon |\nabla z|^2 + \frac{1}{4\epsilon}(z-1)^2 dx + \mu \int_\Omega |u - g|^2 dx$$

where $(s, z, u) \in \left[W^{1,2}(\Omega, [0, 1])\right]^2 \times W^{2,2}(\Omega) = \mathcal{D}(\Omega)$. Here ϵ is the convergence continuous parameter, ξ_ϵ,o_ϵ are infinitesimals and the convergence is intended for $\epsilon \to 0$. To prove Γ-convergence, one has to show that for any $u \in GSBV^2(\Omega), s \equiv 1, z \equiv 1$ the two following properties are verified:

Liminf inequality: for any sequence $(s_\epsilon, z_\epsilon, u_\epsilon)_{\epsilon>0} \subset \mathcal{D}(\Omega)$ that $[L^1(\Omega)]^3$-converges to (s, z, u) it holds that $\mathcal{F} \leq \liminf_{\epsilon\to 0} \mathcal{F}\epsilon(s_\epsilon, z_\epsilon, u_\epsilon)$.

Limsup inequality: there exists a sequence $(s_\epsilon, z_\epsilon, u_\epsilon)_{\epsilon>0} \subset \mathcal{D}(\Omega)$ that $[L^1(\Omega)]^3$-converges to (s, z, u) such that $\limsup_{\epsilon\to 0} \mathcal{F}\epsilon(s_\epsilon, z_\epsilon, u_\epsilon) \leq \mathcal{F}$.

By standard arguments of functional analysis it is possible to prove that for any $\epsilon > 0$ the functional $\mathcal{F}_\epsilon$ always admits a minimizing triplet. Let us denote it by $(s_\epsilon, z_\epsilon, u_\epsilon)$. By sending $\epsilon \to 0$, thanks to the compactness properties of the Γ-convergence, the sequence $(s_\epsilon, z_\epsilon, u_\epsilon)_{\epsilon\geq 0}$ converges in the $[L^1(\Omega)]^3$-norm to triplet (s, z, u) where u is a minimizer of the limit functional $\mathcal{F}$ and $s, z \equiv 1$ almost everywhere over Ω.

The constructive part of the Γ-convergence (Limsup inequality) provides us the tremendous advantage of keeping trace of the discontinuity sets S_u and

$S_u \cup S_{\nabla u}$ via their regular function approximation. For a fixed $\epsilon > 0$, the two discontinuity sets, enjoying the regularity properties of $GSBV^2(\Omega)$ functions, are approximated by s_ϵ and z_ϵ (respectively) using a slicing argument and a coarea-formula for Lipschitz functions [8]. Let S be either S_u or $S_u \cup S_{\nabla u}$ and let us consider a 2-dimensional orthogonal slice of S. The idea is to build a function σ_ϵ that is 0 in a tubular neighborhood of radius b_ϵ of the set S and that tends to 1 smoothly elsewhere. The tubular neighborhood shrinks as $\epsilon \to 0$. Formally the function σ_ϵ is defined as:

$$\sigma_\epsilon := \begin{cases} 0, & (S)_{b_\epsilon} \\ 1 - \eta_\epsilon, & \Omega \backslash (S)_{b_\epsilon + a_\epsilon} \\ h_\epsilon \circ \tau, & \text{elsewhere} \end{cases}$$

where $a_\epsilon, b_\epsilon, \eta_\epsilon$ are infinitesimals as $\epsilon \to 0$, $\tau(y) := dist(y, \overline{S})$ and $(S)_r := y \in \mathbb{R}^2 : dist(y, S) < r$. The function h_ϵ is obtained as the solution of the differential problem $h' = (1-h)/2\epsilon, h(b_\epsilon = 0)$, where $h(b_\epsilon + a_\epsilon) = 1 - \eta_\epsilon$. Exploiting the Schwarz inequality $a^2 + b^2 \geq 2ab$ it is possible to prove that such h_ϵ is energetically optimal in the class of the admissible functions (a general result is given in [220] and used for the approximation of discontinuity sets in [10, 8]). Because of the global minimization of $\mathcal{F}_\epsilon$, the distance term $\mu|u_\epsilon - g|^2$ keeps the function u_ϵ close to g. High values of $|\nabla u_\epsilon|$ (associated to discontinuities of g) and high values of $|\nabla^2 u_\epsilon|$ (associated to crease points of g) force the transition of the functions s_ϵ and z_ϵ from 1 to 0. Elsewhere, the minimization of the two terms containing the differential operators causes the smoothing of g. We remark here the importance of the parameters $\delta, \mu, \alpha, \beta$ that control the ration at which the whole mechanism described before takes place. From the discussion above it follows that, for small values of ϵ the computation of minimizing triplet of 5.2.2 provides u_ϵ an approximation of a real minimizer u of $\mathcal{F}$, and s_ϵ, ζ_ϵ, the functions that map the tubular neighborhoods of the discontinuity sets S_u and $S_{\nabla u} \cup S_u$, respectively. The price to pay for having such nice outputs is computational complexity.

The numerical approach for the minimization of $\mathcal{F}_\epsilon$ exploits a discrete version of the functional. The rectangular domain Ω is discretized by a lattice of points $\Lambda = \{(it_x, jt_y); i = 1, \ldots, N, j = 1, \ldots, M\}$ with step size t_x and t_y on the x y directions, respectively. In order to take into account boundary conditions, this lattice can be viewed as a subset of an enlarged lattice $\overline{\Lambda}$. We denote the

set of points on the frame outside of Ω by $\mathcal{B} = \overline{\Lambda} - \Lambda$. By using the standard representation of grey-scale images as matrices, the values g_{ij} of the given image g are defined only on the grid points (it_x, it_y) of Λ. The approximate values of the functions s, z, u at the grid points of $\overline{\Lambda}$ are denoted by s_{ij}, z_{ij}, u_{ij} assuming zero boundary conditions on $\mathcal{B}$ (i.e. $s_{ij} = z_{ij} = u_{ij} = 0$, $(i, j \in \mathcal{B})$). By using a column-wise ordering for the elements of these matrices, the image matrix is denoted also by the NM vector $\mathbf{g}$ while the vectors $\mathbf{s}_{\overline{\Lambda}}$, $\mathbf{z}_{\overline{\Lambda}}$,$\mathbf{u}_{\overline{\Lambda}}$, denote the entries of the function s, z, u at the grid points of $\overline{\Lambda}$; the discrete functional depends only on the sub-vectors, denoted in the following with $\mathbf{s}, \mathbf{z}, \mathbf{u}$, whose NM entries are the approximate values of s, z, u at the points of Λ; the others entries of $\mathbf{s}_{\overline{\Lambda}}$, $\mathbf{z}_{\overline{\Lambda}}$,$\mathbf{z}_{\overline{\Lambda}}$ are their boundary values which interleave the elements of $\mathbf{s}, \mathbf{z}, \mathbf{u}$ in $\mathbf{s}_{\overline{\Lambda}}$, $\mathbf{z}_{\overline{\Lambda}}$,$\mathbf{z}_{\overline{\Lambda}}$ and form three other sub-vectors denoted with $\mathbf{s}_{\mathcal{B}}$, $\mathbf{z}_{\mathcal{B}}$,$\mathbf{z}_{\mathcal{B}}$.

The first and second order differential operators appearing in the functional can be approximated via finite difference schemes as follows

$$
\begin{aligned}
&\partial_x v_{ij} := \tfrac{v_{i+1,j} - v_{i,j}}{t_x} \\
&\partial_y v_{ij} := \tfrac{v_{i,j+1} - v_{i,j}}{t_y} \\
&\partial_{xx} v_{ij} := \tfrac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{t_x^2} \\
&\partial_{yy} v_{ij} := \tfrac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{t_y^2} \\
&\partial_{xy} v_{ij} := \tfrac{1}{t_y} \left(\tfrac{v_{i+1,j+1} - v_{i,j+1}}{t_x} - \tfrac{v_{i+1,j} - v_{i,j}}{t_x} \right)
\end{aligned}
$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, M$. Moreover, by taking into account all contributions affecting $\mathbf{s}, \mathbf{z}, \mathbf{u}$ from boundaries, by a simple $2-D$ composite rectangular rule, we obtain the following discrete form of the functional (5.2.2).

$$
\begin{aligned}
F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) :=& t_x t_y \left\{ \delta \sum_{i=0}^{N+1} \sum_{j=0}^{M+1} z_{i,j}^2 \left(\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{t_x^2} \right)^2 + \left(\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{t_y^2} \right)^2 \right) \right. \\
&\left. + 2\delta \sum_{i=0}^{N} \sum_{j=0}^{M} z_{i,j}^2 \left(\frac{u_{i+1,j+1} - u_{i,j+1}}{t_y t_x} - \frac{u_{i+1,j} - u_{i,j}}{t_y t_x} \right)^2 \right.
\end{aligned}
$$

$$
\begin{aligned}
&+\xi_\epsilon \textstyle\sum_{i=0}^{N} \sum_{j=0}^{M} \left(s_{i,j}^2 + o_\epsilon \right) \left(\left(\frac{u_{i+1,j} - u_{i,j}}{t_x} \right)^2 + \left(\frac{u_{i,j+1} - u_{i,j}}{t_y} \right)^2 \right) \\
&+(\alpha - \beta) \left[\epsilon \textstyle\sum_{i=0}^{N} \sum_{j=0}^{M} \left(\left(\frac{s_{i+1,j} - s_{i,j}}{t_x} \right)^2 + \left(\frac{s_{i,j+1} - s_{i,j}}{t_y} \right)^2 \right) + \frac{1}{4\epsilon} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(s_{i,j} - 1 \right)^2 \right]
\end{aligned}
$$

$$+\beta\left[\epsilon\sum_{i=0}^{N}\sum_{j=0}^{M}\left(\left(\frac{z_{i+1,j}-z_{i,j}}{t_x}\right)^2+\left(\frac{z_{i,j+1}-z_{i,j}}{t_y}\right)^2\right)+\frac{1}{4\epsilon}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(z_{i,j}-1\right)^2\right]$$
$$+\mu\sum_{i=1}^{N}\sum_{j=1}^{M}\left(u_{i,j}-g_{i,j}\right)^2\}$$

with $2\beta \geq \alpha \geq \beta > 0$ and $\delta, \mu, \epsilon > 0.\xi_\epsilon, o_\epsilon, \geq 0$. In order to write in compact matrix form the functional 2.1.2.5, we introduce the square matrices $\mathbf{A}_k^1, \mathbf{A}_k^2$ of order $K+4$, representing first and second order discrete operators respectively:

$$\mathbf{A}_K^1 := \begin{Bmatrix} 0 & 0 & & & & & \\ & -1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 1 & & \\ & & & & 0 & 0 & \\ & & & & & 0 & \end{Bmatrix} \mathbf{A}_K^2 := \begin{Bmatrix} 0 & 0 & & & & & \\ 1 & -2 & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 0 & 0 & \end{Bmatrix}$$

By denoting by $\mathbf{I}_K$ and $\mathbf{0}_K$ the identity and the null matrices of order K respectively and by $\otimes$ the Kronecker product, we can write the first and the second order discrete operators in the following way:

$$\begin{aligned}
\mathbf{D}_x &:= \tfrac{1}{t_x}\mathbf{W}_M \otimes \mathbf{A}_N^1; \quad \mathbf{D}_y := \tfrac{1}{t_y}\mathbf{A}_M^1 \otimes \mathbf{W}_N \\
\mathbf{D}_{xx} &:= \tfrac{1}{t_x^2}\mathbf{W}_M \otimes \mathbf{A}_N^2; \quad \mathbf{D}_{yy} := \tfrac{1}{t_y^2}\mathbf{A}_M^2 \otimes \mathbf{W}_N \\
\mathbf{D}_{xy} &:= \tfrac{1}{t_x t_y}\left(\mathbf{A}_M^1 \otimes \mathbf{J}_N\right)\left(\mathbf{J}_M \otimes \mathbf{A}_N^1\right) = \tfrac{1}{t_x t_y}\left(\mathbf{J}_M \otimes \mathbf{A}_N^1\right)\left(\mathbf{A}_M^1 \otimes \mathbf{J}_N\right)
\end{aligned}$$

with $\mathbf{W}_M = \mathrm{diag}(\mathbf{0}_2, \mathbf{I}_M, \mathbf{0}_2)$, $\mathbf{W}_N = \mathrm{diag}(\mathbf{0}_2, \mathbf{I}_N, \mathbf{0}_2)$, $\mathbf{J}_M = \mathrm{diag}(\mathbf{0}_1, \mathbf{I}_{M+1}, \mathbf{0}_2)$ and $\mathbf{J}_N = \mathrm{diag}(\mathbf{0}_1, \mathbf{I}_{N+1}, \mathbf{0}_2)$.

In the following, for any generic $(N+4)(M+4)$ matrix operator $\mathbf{D}$, let us denote by $\mathbf{D}(:, \Lambda)$ and $\mathbf{D}(:, \mathcal{B})$ the sub-matrices of $\mathbf{D}$ given by the columns with indices corresponding to points of Λ and $\mathcal{B}$ respectively, ordered in a column-wise way. Furthermore, given a generic vector $\mathbf{v}$, let us denote by $\mathbf{R}_\mathbf{v}$ the diagonal matrix with diagonal entries equal to the elements of $\mathbf{v}$ i.e. $\mathbf{R}_\mathbf{v} = \mathrm{diag}(\mathbf{v})$. We also denote by $\mathbf{v}^2$ the vector of the squared entries of $\mathbf{v}$, i.e. $(\mathbf{v}^2)_i = (\mathbf{v}_i)^2$ and $\mathbf{1} = (1, 1, \ldots, 1)^T$.

Using this notation, the discrete functional can be written as follows:

$$F_\epsilon(\mathbf{s},\mathbf{z},\mathbf{u}) := t_x t_y \left\{ \delta \mathbf{u}_{\overline{\Lambda}}^T \left(\mathbf{D}_{xx}^T \mathbf{R}_{z_{\overline{\Lambda}}^2} \mathbf{D}_{xx} + \mathbf{D}_{yy}^T \mathbf{R}_{z_{\overline{\Lambda}}^2} \mathbf{D}_{yy} + 2\mathbf{D}_{xy}^T \mathbf{R}_{z_{\overline{\Lambda}}^2} \mathbf{D}_{xy} \right) \mathbf{u}_{\overline{\Lambda}} \right.$$

$$\xi_\epsilon \mathbf{u}_{\overline{\Lambda}}^T \left(\mathbf{D}_x^T \mathbf{R}_{\mathbf{s}_{\overline{\Lambda}}^2 + o_\epsilon} \mathbf{D}_x + \mathbf{D}_y^T \mathbf{R}_{\mathbf{s}_{\overline{\Lambda}}^2 + o_\epsilon} \mathbf{D}_y \right) \mathbf{u}_{\overline{\Lambda}}$$

$$(\alpha - \beta) \left[\epsilon \mathbf{s}_{\overline{\Lambda}}^T \left(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y \right) \mathbf{s}_{\overline{\Lambda}} + \frac{1}{4\epsilon} (\mathbf{s} - \mathbf{1})^T (\mathbf{s} - \mathbf{1}) \right]$$

$$\beta \left[\epsilon \mathbf{z}_{\overline{\Lambda}}^T \left(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y \right) \mathbf{z}_{\overline{\Lambda}} + \frac{1}{4\epsilon} (\mathbf{z} - \mathbf{1})^T (\mathbf{z} - \mathbf{1}) \right]$$

$$\left. + \mu (\mathbf{u} - \mathbf{g})^T (\mathbf{u} - \mathbf{g}) \right\}.$$

Globally this functional is not convex, but it is quadratic with respect to each block of variables $\mathbf{s}, \mathbf{z}, \mathbf{u}$ when the others are fixed: the terms F_ϵ containing $\mathbf{s}$ or $\mathbf{z}$ depend only on $\mathbf{u}$ and, on the other hand, the terms containing $\mathbf{u}$ depend on $\mathbf{s}$ and $\mathbf{z}$. Indeed, by fixing the variable $\mathbf{u}$ or the other two variables $\mathbf{s}$ and $\mathbf{z}$ we can write

$$F_\epsilon(\mathbf{s},\mathbf{z},\mathbf{u}) = t_x t_y \left\{ \frac{1}{2} \left(\mathbf{s}^T \mathbf{z}^T \right) \begin{pmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \mathbf{s} \\ \mathbf{z} \end{pmatrix} - \left(\mathbf{s}^T \mathbf{z}^T \right) \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} + \mathbf{c}_{sz} \right\}$$

$$F_\epsilon(\mathbf{s},\mathbf{z},\mathbf{u}) = t_x t_y \left\{ \frac{1}{2} \mathbf{u}^T \mathbf{A}_3 \mathbf{u} - \mathbf{u}^T \mathbf{b}_3 + \mathbf{c}_u \right\}$$

where $\mathbf{A_1} = \mathbf{A_1}(\mathbf{u})$ $\mathbf{A_2} = \mathbf{A_2}(\mathbf{u})$ $\mathbf{A_3} = \mathbf{A_3}(\mathbf{s}, \mathbf{z})$ and $\mathbf{b_1}$, $\mathbf{b_2}$, $\mathbf{b_3}$ are given by

$$
\begin{aligned}
&\mathbf{A}_1 = 2\xi_\epsilon \mathbf{R}_{|\nabla \mathbf{u}|^2} + 2\epsilon(\alpha-\beta)\left(\mathbf{D}_x(:,\Lambda)^T\mathbf{D}_x(:,\Lambda) + \mathbf{D}_y(:,\Lambda)^T\mathbf{D}_y(:,\Lambda)\right) + \tfrac{\alpha-\beta}{2\epsilon}\mathbf{I}_{NM}\\
&\mathbf{b}_1 = -2\epsilon(\alpha-\beta)\left(\mathbf{D}_x(:,\Lambda)^T\mathbf{D}_x(:,\mathcal{B}) + \mathbf{D}_y(:,\Lambda)^T\mathbf{D}_y(:,\mathcal{B})\right)\mathbf{s}_\mathcal{B} + \tfrac{\alpha-\beta}{2\epsilon}\mathbf{1}\\
&\mathbf{A}_2 = 2\delta \mathbf{R}_{|\nabla^2 \mathbf{u}|^2} + 2\epsilon\beta\left(\mathbf{D}_x(:,\Lambda)^T\mathbf{D}_x(:,\Lambda) + \mathbf{D}_y(:,\Lambda)^T\mathbf{D}_y(:,\Lambda)\right) + \tfrac{\beta}{2\epsilon}\mathbf{I}_{NM}\\
&\mathbf{b}_2 = -2\epsilon\beta\left(\mathbf{D}_x(:,\Lambda)^T\mathbf{D}_x(:,\mathcal{B}) + \mathbf{D}_y(:,\Lambda)^T\mathbf{D}_y(:,\mathcal{B})\right)\mathbf{z}_\mathcal{B} + \tfrac{\beta}{2\epsilon}\mathbf{1}\\
&\mathbf{A}_3 = 2\delta\Big(\mathbf{D}_{xx}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{xx}(:,\Lambda) + \mathbf{D}_{yy}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{yy}(:,\Lambda)\\
&+2\mathbf{D}_{xy}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{xy}(:,\Lambda) + 2\xi_\epsilon\mathbf{D}_x(:,\Lambda)^T\mathbf{R}_{\mathbf{s}^2_\Lambda+o_\epsilon}\mathbf{D}_x(:,\Lambda)+\\
&\mathbf{D}_y(:,\Lambda)^T\mathbf{R}_{\mathbf{s}^2_\Lambda+o_\epsilon}\mathbf{D}_y(:,\Lambda) + 2\mu\mathbf{I}_{NM},\\
&\mathbf{b}_3 = -2\delta\Big(\mathbf{D}_{xx}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{xx}(:,\mathcal{B}) + \mathbf{D}_{yy}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{yy}(:,\mathcal{B}\\
&+2\mathbf{D}_{xy}(:,\Lambda)^T\mathbf{R}_{\mathbf{z}^2_\Lambda}\mathbf{D}_{xy}(:,\mathcal{B}) + 2\xi_\epsilon\mathbf{D}_x(:,\Lambda)^T\mathbf{R}_{\mathbf{s}^2_\Lambda+o_\epsilon}\mathbf{D}_x(:,\mathcal{B})+\\
&\mathbf{D}_y(:,\Lambda)^T\mathbf{R}_{\mathbf{s}^2_\Lambda+o_\epsilon}\mathbf{D}_y(:,\mathcal{B})\mathbf{u}_\mathcal{B} + 2\mu\mathbf{g}.
\end{aligned}
$$

Vectors $\mathbf{c}_{sz}$ and $\mathbf{c}_u$ are constant, thus irrelevant for the minimization.

Remark1. We observe that if $\alpha = \beta$ and $\xi_\epsilon = 0$, the functional does not depend any more on the block variable $\mathbf{s}$. The features and the numerical treatment of this reduced version of F_ϵ are similar to the ones of the general case, with the only difference that F_ϵ is quadratic with respect to the block variable $\mathbf{z}$ when $\mathbf{u}$ is fixed, since $A_1 = 0$, $b_1 = 0$. Thus, since the setting $\alpha = \beta$ and $\zeta_\epsilon = 0$ is a special case, in the following we address the general formulation, assuming $\alpha > \beta$ and $\zeta_\epsilon > 0$. The functional F_ϵ 2.1.2.5 has the following properties:

P1. $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$ is continuously differentiable;

P2. The matrices $\mathbf{A}_i, i = 1, 2, 3$ are sparse and structured: $\mathbf{A}_1$ and $\mathbf{A}_2$ are block tridiagonal matrices, where the diagonal blocks are tridiagonal and the off-diagonal blocks are diagonal; $\mathbf{A}_3$ is a block five matrix, with at most 13 nonzero entries for each row;

P3. In view of the terms $\frac{\alpha-\beta}{2\epsilon}\mathbf{I}$, $\frac{\beta}{\epsilon}\mathbf{I}$ and $2\mu\mathbf{I}$ the matrices $\mathbf{A}_i, i = 1, 2, 3$ are symmetric and positive definite and their minimum eigenvalues $\lambda_{min}(\mathbf{A}_i)$ are below bounded by $\frac{\alpha-\beta}{2\epsilon}$, $\frac{\beta}{2\epsilon}$ and 2μ, respectively;

P4. In view of the positive definiteness of matrices $\mathbf{A}_i$, $i = 1, 2, 3$ $F_\epsilon(s, z, u)$ is a quadratic and strongly convex with respect to each block component $\mathbf{s}, \mathbf{z}, \mathbf{u}$ when the others are left fixed;

P5. F_ϵ is coercive in $\mathbb{R}^{3NM}$; thus the level sets of F_ϵ are compact;

P6. On a given level set, the matrices $\mathbf{A}_i, i = 1, 2, 3$ have bounded positive eigenvalues;

P7. On given level set, the gradient of F_ϵ is Lipschitz continuous: given a level set $\mathcal{L}_\eta = \{(\mathbf{s}, \mathbf{z}, \mathbf{u}) : F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) \leq \eta\}$ there exists a constant $M \geq 0$ such that $||\nabla F_\epsilon(\mathbf{s}_1, \mathbf{z}_1, \mathbf{u}_1) - \nabla F_\epsilon(\mathbf{s}_2, \mathbf{z}_2, \mathbf{u}_2)|| \leq M||(\mathbf{s}_1, \mathbf{z}_1, \mathbf{u}_1) - (\mathbf{s}_2, \mathbf{z}_2, \mathbf{u}_2)||$ for any $(\mathbf{s}_1, \mathbf{z}_1, \mathbf{u}_1), (\mathbf{s}_2, \mathbf{z}_2, \mathbf{u}_2) \in \mathcal{L}_\eta$

P8. The functional F_ϵ is a polynomial in $\mathbf{s}, \mathbf{z}, \mathbf{u}$ and consequently, it is a semi-algebraic function, that satisfies the Kurdyka-Lojasiewicz (KL) property on its domain (see [18] and reference therein).

In the following, for notation convenience, a generic point $(\mathbf{s}, \mathbf{z}, \mathbf{u})$ in $\mathbb{R}^{3NM}$ is represented either by $\mathbf{y}$ or $\mathbf{x}$. When $\mathbf{y}$ is used, the variables are grouped in blocks according the simple correspondence: $\mathbf{y}_1 = \mathbf{s}, \mathbf{y}_2 = \mathbf{z}$ and $\mathbf{y}_3 = \mathbf{u}$; on the other hand, we refer to $\mathbf{x}$ when block decomposition is based on the spatial subdivision of Λ. The functional restricted to a block variable $\mathbf{v}_i$ is denoted by $f_{\mathbf{v}_i} = F_\epsilon(\ldots, \mathbf{v}_i, \ldots)$; the gradient of F_ϵ at $\tilde{\mathbf{v}}$ with respect to the block $\mathbf{v}_i$ is denoted by $\nabla_{\mathbf{v}_i} F_\epsilon(\tilde{\mathbf{v}})$ and $\nabla_{\mathbf{v}_i} F_\epsilon(\tilde{\mathbf{v}}) = \nabla f_{\mathbf{v}_i}(\tilde{\mathbf{v}})$

Sequential Approach In view of the remark that the discrete approximation of 2.1.2.5 is a polynomial function, satisfying the Kurdyka-Lojasiewicz property, we obtain the convergence of the sequence generated by BCDA to a critical point. BCDA is a version of block coordinate descent algorithm [155], especially tailored to exploit the features of the functional $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$. In 1

Algorithm 1: BCDA

Step 0 Given
$\mathbf{s}^0$,$\mathbf{z}^0$,$\mathbf{u}^0, 0 < \rho_i < 1, \gamma_i \in (0, 2(1-\rho_i)], i = 1, 2, 3$ and an exit tolerance θ_{outer};

Step 1 k=0

Step 2 Inexact minimization with respect to $\mathbf{s}$ and $\mathbf{z}$:

1. compute the search directions $\mathbf{d}_1^k$ and $\mathbf{d}_2^k$;
2. compute $\alpha_1^k = \gamma_1 \frac{(\mathbf{b}_1 - \mathbf{A}_1^k \mathbf{s}^k)^T \mathbf{d}_1^k}{\mathbf{d}_1^{kT} \mathbf{A}_1^k \mathbf{d}_1^k}$, $\alpha_2^k = \gamma_2 \frac{(\mathbf{b}_2 - \mathbf{A}_2^k \mathbf{z}^k)^T \mathbf{d}_2^k}{\mathbf{d}_2^{kT} \mathbf{A}_2^k \mathbf{d}_2^k}$
3. update $\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha_1^k d_1^k$; $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha_2^k d_2^k$.

Step 3 Inexact minimization with respect to $\mathbf{u}$

1. compute the search directions $\mathbf{d}_3^k$;
2. compute $\alpha_3^k = \gamma_3 \frac{(\mathbf{b}_3 - \mathbf{A}_3^k \mathbf{s}^k)^T \mathbf{d}_3^k}{\mathbf{d}_3^{kT} \mathbf{A}_3^k \mathbf{d}_3^k}$
3. update $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha_3^k \mathbf{d}_3^k$

Step 4 if $(F_\epsilon(\mathbf{y}^k) - F_\epsilon(\mathbf{y}^{k+1})) \leq \theta_{outer} F_\epsilon(\mathbf{y}^{k+1})$ then stop; else $k = k + 1$ and go to Step 2.

Starting from an initial vector $\mathbf{y}^0 = (\mathbf{s}^0, \mathbf{z}^0, \mathbf{u}^0)$, the basic idea of the method is to cyclically determine for each block variable $\mathbf{y}_i$ a descent direction $\mathbf{d}_i$ by few iterations of a preconditioned conjugate gradient (PCG) method applied to the linear system $\mathbf{A}_i^k \mathbf{d}_i = \mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k = -\nabla_{\mathbf{y}_i} F_\epsilon(\mathbf{y}^k)$, with $\mathbf{A}_i^k \equiv \mathbf{A}_i^k(\mathbf{y}^k)$. In view of the property **P4** of the objective function, the step-lengths along the computed descent directions $\mathbf{d}_i^k$ at the iteration k can be determined without having to use an Armijo-type procedure to ensure a sufficient decrease of the objective function. Indeed when the step size α_i^k is given by the following rule

$$\alpha_i^k := \gamma_i \frac{\left(\mathbf{b}_i - \mathbf{A}_\mathbf{i}^k \mathbf{y}_i^k\right)^T \mathbf{d}_i^k}{\mathbf{d}_i^k \mathbf{A}_i^k \mathbf{d}_i^k} = \gamma_i \frac{-\nabla_{\mathbf{y}_i} F_\epsilon \left(\mathbf{y}^k\right)^T \mathbf{d}_i^k}{\mathbf{d}_i^k \mathbf{A}_i^k \mathbf{d}_i^k},$$

with $0 \leq \gamma_i \leq 2(1-\rho_i), 0 \leq \rho_i \leq 1$ a sufficient decrease for the objective function restricted to the block variable $\mathbf{y}_i$ is assured:

$$f_{\mathbf{y}_i}\left(\mathbf{y}_i^k + \alpha_i^k \mathbf{d}_i^k\right) \leq f_{\mathbf{y}_i}\left(\mathbf{y}_i^k\right) + \rho_i \alpha_i^k \nabla f_{\mathbf{y}_i}\left(\mathbf{y}_i^k\right)^T \mathbf{d}_i^k.$$

In particular, for $\gamma_i = 1$ we obtain the exact one-dimensional minimizer of the strongly convex quadratic function along the direction $\mathbf{d}_i^k$ (2.1.2.5 holds for $\rho_i \leq \frac{1}{2}$). In view of **P5**, the level set $\mathcal{L}_{F_\epsilon^0} = \{(\mathbf{s}, \mathbf{z}, \mathbf{u}) : F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) \leq F_\epsilon^0 = F_\epsilon(\mathbf{s}^0, \mathbf{z}^0, \mathbf{u}^0)\}$ is a compact subset of $\mathbb{R}^{3MN}$. Thus, for $(\mathbf{s}, \mathbf{z}, \mathbf{u}) \in \mathcal{L}_{F_\epsilon^0}$, the eigenvalues $\lambda_j(\mathbf{A}_i^k)$ of the matrices $\mathbf{A}_i^k, i = 1, 2, 3k \geq 0$ are bounded by positive constants (**P6**):

$$0 \leq \lambda_m \leq \lambda_j(\mathbf{A}_i^k) \leq \lambda_M, k \geq 0, j = 1, \ldots, NM$$

where $\lambda_m = \min\{2\mu, \frac{\alpha-\beta}{2\epsilon}, \frac{\beta}{2\epsilon}\}$ and their condition numbers have above been bounded by a positive constant $L \leq \frac{\lambda_M}{\lambda_m}$.

It is easy to prove that the directions $\mathbf{d}_i^k, i = 1, 2, 3$ computed at any k-iteration of BCDA by PCG with a suitable stopping rule are gradient related search directions. Consequently, since BCDA is a special version of the Algorithm 1 in [155], Theorem 7.1 in [155] states that $\nabla F_\epsilon(\mathbf{s}^k, \mathbf{z}^k, (\mathbf{u}^k) \to 0$ as $k \to \infty$ and there exists at least a limit point of $\{\mathbf{s}^k, \mathbf{z}^k, (\mathbf{u}^k\}$ in $\mathcal{L}_{F_\epsilon^0}$ that is a stationery point of F_ϵ. The following proposition resumes the features of $\mathbf{d}_i^k, i = 1, 2, 3, k \geq 0$.

Theorem 1. *Let assume that* $\nabla_{\mathbf{y}_i} F_\epsilon(\mathbf{y}^k) = \mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i \neq 0$ *for* $k \geq 0$. *Let consider the PCG method applied to the symmetric positive definitive system* $\mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k = \mathbf{A}_i^k \mathbf{d}_i$. *Let* $\mathbf{d}_i^h$ *be the vector satisfying at the h-iteration of the PCG method the stopping rule*

$$||\mathbf{r}^h|| \leq \eta_i^k ||\mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i|| \text{ with } \eta_i^k \leq \frac{c}{\sqrt{K(\mathbf{A}_i^k)}},$$

where $\mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{A}_i^k \mathbf{d}_i^h$ *is the residual of the linear system,* $K(\mathbf{A}_i^k)$ *is the spectral condition number of* $\mathbf{A}_i^k$ *and* $0 \leq c \leq 1$. *Thus the direction* $\mathbf{d}_i^k := \mathbf{d}_i^h$ *is a gradient related search direction i.e.*

$$\begin{aligned} \frac{\nabla_{\mathbf{y}_i} F_\epsilon \left(\mathbf{y}^k\right)^T \mathbf{d}_i^k}{\left\|\mathbf{d}_i^k\right\|} &\leq \frac{1}{2\left(1+\eta_i^k\right)} \left(\left(\eta_i^k\right)^2 - \frac{1}{K\left(\mathbf{A}_i^k\right)} \right) \left\|\nabla_{\mathbf{y}_i F_\epsilon} \left(\mathbf{y}^k\right)\right\| \\ &\leq \frac{c^2 - 1}{4L} \left\|\nabla_{\mathbf{y}_i} F_\epsilon \left(\mathbf{y}^k\right)\right\| \end{aligned}$$

with $c^2 - 1 < 0$

since F_ϵ is a KL function (**P8**), we can obtain also the convergence of the sequence of iterates $\{\mathbf{y}^k\}$ generated by BCDA to some critical point of F_ϵ in $\mathcal{L}_{F_\epsilon^0}$. Indeed, the analysis of an abstract descent algorithm for a KL function and the

convergence results of Theorem 2.9 in [18] enables us to obtain similar results for BCDA. In particular, in view of the continuity of F_ϵ, we have that Theorem 2.9 in [18] assures that the sequence $\{\mathbf{y}^k\}$ generated by BCDA method converges to some critical point of F_ϵ in $\mathcal{L}_{F_\epsilon^0}$ if the following conditions hold for $\{\mathbf{y}^k\}$:

- sufficient decrease condition

$$F_\epsilon\left(\mathbf{y}^{k+1}\right) + C_1 \left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\|^2 \leq F_\epsilon\left(\mathbf{y}^k\right)$$

- relative error condition

$$\left\|\nabla F_\epsilon(\mathbf{s}^{k+1}, \mathbf{z}^{k+1}, \mathbf{u}^{k+1})\right\| \leq C_2 \left\|\mathbf{y}^{k+1} - \mathbf{y}^{k+1}\right\|$$

where C_1 and C_2 are fixed positive constants. The following two propositions verify the validity of the two conditions 2.1.2.5 and 2.1.2.5 for a generic iteration of BCDA.

Theorem 2. *At any k-iteration of BCDA, $k \geq 0$, while $\nabla F_\epsilon(y^k) \neq 0$ we have that 2.1.2.5 holds with $0 < C_1 \leq \lambda_m min_i \frac{\rho_i}{\gamma_i}$*

In the proof of the following proposition, it is crucial to observe that $\nabla_s F_\epsilon(y) = \nabla_s F_\epsilon(s,u)$, while $\nabla_z F_\epsilon(y) = \nabla_z F_\epsilon(z,u)$ and $\nabla_u F_\epsilon(y) = \nabla_u F_\epsilon(s,z,u)$. Furthermore, we recall that the gradient of F_ϵ on the level set $\mathcal{L}_{F_\epsilon^0}$ is M-Lipschitz continuous (**P7**)

Theorem 3. *At any k-iteration of BCDA, $k \geq 0$, while $\nabla F_\epsilon(\mathbf{y}^k) \neq 0$, we have that 2.1.2.5 holds with $C_2 \geq \frac{\sqrt{48}\lambda_M}{(1-c^2)\min_i \gamma_i} + 4M$*

Because of theorem 2.9 in [18], the two previous propositions enable to obtain the following convergence result.

Theorem 4. *Given a starting point $\mathbf{y}_0$, the sequence $\{\mathbf{y}^k\}$ generated by BCDA converges to some critical point of F_ϵ in $\mathcal{L}_{F_\epsilon^0}$. Moreover the sequence $\{\mathbf{y}^k\}$ has finite length, i.e. $\sum_{k=0}^{\infty} \left\|\mathbf{y}^{k+1} - \mathbf{y}^k\right\| < \infty$*

Since BCDA has the same properties of an abstract descent method satisfying 2.1.2.5, 2.1.2.5 and the discrete BZ function is a polynomial (that is a real analytic function), results about the rate convergence of BCDA are obtained in Theorem 4

in [126]. Indeed, in this case, the desingularizing function related to the Kurdyka-Lojasiewicz at a critical point of F_ϵ is of the form $\phi(t) = \frac{c}{\sigma}t^\sigma$, with $C > 0$ and $\sigma \in (0, 1]$; the convergence can be obtained in a finite number of step for $\sigma = 1$, while for $\sigma \in (\frac{1}{2}, 1)$ we have exponential convergence and for $\sigma \in (0, \frac{1}{2})$ polynomial convergence. We conclude the section, by stating in the following proposition a feature of BCDA method, useful for the next section.

Theorem 5. *Let assume that K iterations of BCDA are executed, with $K \leq \overline{K}$, thus we have*

$$\frac{C_1}{\overline{K}} \left\|\mathbf{y}^k - \mathbf{y}^0\right\|^2 \leq \frac{C_1}{K} \left\|\mathbf{y}^k - \mathbf{y}^0\right\|^2 \leq F_\epsilon(\mathbf{y}^0) - F_\epsilon(\mathbf{y}^k)$$

Parallel Approach We are interested in parallel algorithms for solving the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

where f is a differentiable function from the n-dimensional real space $\mathbb{R}^n$ into $\mathbb{R}$. The basic idea behind our approach is to assign a portion of the gradient ∇f of f to one of k processors, let each processor perform one or more steps of a serial algorithm on its portion of the gradient, and then synchronize the processors eventually. The synchronization consists of taking a strong convex combination of the k points found by the k processors when f is convex. For non convex f, the best point found by the k processors can be taken, or any other point with a lower value of f will work. The algorithms typically consist of a direction choice followed by a stepsize choice. The combined direction-stepsize choice generates a decrease in the objective function that forces the eventual satisfaction of an optimality condition, namely, the vanishing of the gradient. Direction choices include descent directions, Newton, quasi-Newton and conjugate directions. Stepsize choice along the chosen direction include minimization, finding the first stationary point, interval stepsize, the Armijo stepsize and others. Related algorithms, wherein the objective function is sequentially minimized with respect to certain variables, include the serial algorithm proposed by Warga for a strictly convex function in each block of variables and in which the function is sequentially minimized for each block of variables, and the coordinate descent methods of Tseng and Luo and Tseng .

In this section, we address the minimization problem of the discrete functional F_ϵ by subdiving the lattice Λ into p tiles $T_1, \ldots, T_p$ with $T_i \cap T_j = \emptyset, i \neq j$. This subdivision leads a partition of the variable $\mathbf{x} = (\mathbf{s}, \mathbf{z}, \mathbf{u})$ into p blocks $\mathbf{x}_{T_1}, \mathbf{x}_{T_2}, \ldots, \mathbf{x}_{T_p}$, with $\mathbf{x}_{T_j} \in \mathbb{R}^{n_j}, j = 1 \ldots p, \sum_{j=1}^{p} n_j = NM$. Here each block of variables $\mathbf{x}_{T_j}$ includes the approximations of the functions $s_{i,j}, z_{i,j}, u_{i,j}$ related to the points of Λ belonging to $n_j = N_j \times M_j$ tile T_j, denoted by $\mathbf{s}_{T_j}, \mathbf{z}_{T_j}, \mathbf{u}_{T_j}$, with $\sum_{j=1}^{p} N_j = N, \sum_{j=1}^{p} M_j = M$. In addition to this partition of Λ, we consider a further partition of Λ into partially overlapping p tiles S_j of size $(N_j+\nu)\times(M_j+\nu)$ where S_j is the tile T_j with an outer frame of ν rows and columns of pixels, that is ν is the number of rows/columns of overlapping pixels and $T_j \subset S_j$. Denoting by $\mathbf{x}_{S_j}$ the variables related to S_j, we observe that the vector of variables $\mathbf{x}$ can be considered as the union of $\mathbf{x}_{S_j}$ and $\mathbf{x}_{\Lambda \setminus S_j}$ for any $j = 1 \ldots p$, and that $\mathbf{x}_{S_j}$ is also the union of $\mathbf{x}_{T_j} = \mathbf{s}_{T_j}, \mathbf{z}_{T_j}, \mathbf{u}_{T_j}$ related to the tile T_j and $\mathbf{x}_{B_j} = \mathbf{s}_{B_j}, \mathbf{z}_{B_j}, \mathbf{u}_{B_j}$ related to the frame $B_j = S_j \setminus T_j$. In the following we describe a parallel version of the BCDA scheme, which exploits a decomposition of the image domain in overlapping tiles and enables to address the segmentation of large images by solving a sequence of independent smaller problems, without any necessity of post-processing procedure. After the introduction of the method and the analysis of its theoretical and converging properties, details of our practical parallel implementation are given.

Inspired by [209] and [155] we proposed the parallel algorithm detailed in algorithm 2. We consider the same tiling partitions (with and without overlapping) already mentioned previously. In this scheme, starting from an initial point $\mathbf{x^0}$, at any ℓ iteration, we compute for each tile $S_j, j = 1, \ldots, p$, an inexact minimum point of the objective function restricted to the variables related to S_j, with boundary conditions given by the values of the previous iterate $\mathbf{x}^\ell$ on the frame of S_j.

Then, from the inexact computed solution, only the entries corresponding to the tile T_j are extracted, neglecting the value on B_j. Then, the new iterate is updated by a *connection rule*, ensuring that the value of F_ϵ does not increase. In particular, exploiting the local features of the functional, the value of F_ϵ at the point $\widetilde{\mathbf{m}} = (\overline{\mathbf{x_1}}, \overline{\mathbf{x_2}}, \ldots, \overline{\mathbf{x_p}})$ is computed and, if $F_\epsilon(\widetilde{\mathbf{m}}) \leq F_\epsilon(\mathbf{m^j})$ for all $j = 1, \ldots, p$ we set $\mathbf{x}^{\ell+1} = \widetilde{\mathbf{m}}$. Otherwise, the new iterate is given by the rule $\mathbf{x}^{\ell+1} = \text{argmin}\{F_\epsilon(\mathbf{m^1}), \ldots, F_\epsilon(\mathbf{m^p})\}$. This connection rule and a suitable implementation of the BCDA method to obtain an inexact minimum for each inner

Algorithm 2: Parallel BCDA

Step 0 : Given $\mathbf{x}^0$, the partitions $\{T_1, \ldots, T_p\}$ and $\{S_1, \ldots, S_p\}$ of Λ such that $T_j \subset S_j, B_j = S_j - T_j, j = 1, \ldots, p$ and $\{\theta_\ell\}$ such that $\underline{\theta} < \theta_\ell \leq \overline{\theta}, \ell \geq 0$ and an exit tolerance θ_{outer};

Step 1 : $\ell = 0$; fix the parameters $\gamma_i, \rho_i, i = 1, 2, 3$ and $\overline{K}$ for BCDA.

Step 2 : for $j = 1, \ldots, p$ if $\nabla_{\mathbf{x}_{T_j}} F_\epsilon(\mathbf{x}^\ell) \neq 0$ then compute $\mathbf{m}^j = (\mathbf{x}_1^\ell, \ldots, \mathbf{x}_{j-1}^\ell, \overline{\mathbf{x}_j}, \mathbf{x}_{j+1}^\ell, \ldots, \mathbf{x}_p^\ell)$ as follows:

1. set $\mathbf{x}_{S_j}^0 = (\mathbf{x}^\ell)|_{S_j}$, k=-1;
2. repeat
   (a) $k = k+1$; compute $\mathbf{x}_{S_j}^{k+1}$ by step of BCDA; extract $\mathbf{x}_{T_j}^{k+1}$; set $\overline{\mathbf{x}_j} = \mathbf{x}_{T_j}^{k+1}$;
   (b) if $f_{\mathbf{x}_{S_j}}(\mathbf{x}_{T_j}^k, \mathbf{x}_{B_j}^0) - f_{\mathbf{x}_{S_j}}(\mathbf{x}_{T_j}^{k+1}, \mathbf{x}_{B_j}^0) < C_1 \left\| \mathbf{x}_{T_j}^k - \mathbf{x}_{T_j}^{k+1} \right\|^2$ then redefine $\overline{\mathbf{x}_j} = \mathbf{x}_{T_j}^k$ exit next j; end

   until $\left\| \nabla f_{\mathbf{x}_{S_j}} \right\| \leq \theta_\ell \left\| \mathbf{x}_{S_j}^{k+1} - \mathbf{x}_{S_j}^\ell \right\|$

else
$\mathbf{m}^j = \mathbf{x}^\ell$;
end

Step 3 define the new iterate $\mathbf{x}^{\ell+1}$

1. compute $F_\epsilon(\tilde{\mathbf{m}})$ where $\tilde{\mathbf{m}} = (\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_p)$
2. update $\mathbf{x}^{\ell+1} = \text{argmin}\{F_\epsilon(\tilde{\mathbf{m}}), F_\epsilon(\mathbf{m}^1), \ldots, F_\epsilon(\mathbf{m}^p)\}$

Step 4 if $(F_\epsilon(\mathbf{y}^\ell) - F_\epsilon(\mathbf{y}^{\ell+1})) \leq \theta_{outer} F_\epsilon(\mathbf{y}^{\ell+1})$ then stop; else $\ell = \ell + 1$ and go to Step 2.

subproblem enable to apply Theorem 3.5 in [209], which guarantees the stationarity of any limit point of the sequence $\{\mathbf{x}^\ell\}$ and, since $\mathcal{L}_{F_\epsilon^0}$ is compact, also ensures that a limit point of $\{\mathbf{x}^\ell\}$ exists and the gradients sequence enjoys the following property: $\nabla F_\epsilon(\mathbf{x}^\ell) \to 0$ as $\ell \to \infty$. Indeed it is simple to verify that at any ℓ-iteration of the algorithm the following conditions hold for $\ell \geq 0$ and a positive constant C_3:

$$F_\epsilon(\mathbf{x}^{\ell+1}) \leq F_\epsilon(\mathbf{m}^j) \leq F_\epsilon(\mathbf{x}^\ell) \quad j = 1, \ldots, p$$

$$C_3 ||\nabla_{\mathbf{x}_{T_j}} F_\epsilon(\mathbf{x}^\ell)||^2 \leq F_\epsilon(\mathbf{x}^\ell) - F_\epsilon(\mathbf{x}^{\ell+1}) \quad j = 1, \ldots, p$$

In particular, condition 2.1.2.5 is preserved when at the first step of BCDA a sufficient decrease of the objective $f_{\mathbf{x}_{S_j}}$ is assured with respect to the variables $\mathbf{x}_{T_j}$. In order to obtain this decrease, at the first step of BCDA, the subvectors $(\mathbf{d}_i^0)_{|T_j}$ of the computed descent directions $\mathbf{d}_i^0, i = 1, 2, 3$, restricted to entries of T_j, have to be gradient related to $g_1 \equiv \nabla_{\mathbf{s}_{T_j}} f_{\mathbf{x}_j}(\mathbf{s}_{T_j}^0, \mathbf{z}_{T_j}^0, \mathbf{u}_{T_j}^0, \mathbf{x}_{B_j}^0), g_2 \equiv \nabla_{\mathbf{z}_{T_j}} f_{\mathbf{x}_j}(\mathbf{s}_{T_j}^1, \mathbf{z}_{T_j}^0, \mathbf{u}_{T_j}^0, \mathbf{x}_{B_j}^0), g_3 \equiv \nabla_{\mathbf{u}_{T_j}} f_{\mathbf{x}_j}(\mathbf{s}_{T_j}^1, \mathbf{z}_{T_j}^1, \mathbf{u}_{T_j}^0, \mathbf{x}_{B_j}^0)$ respectively; using an argument similar to the one in the proof of Theorem 1, this condition is verified if the inner PCG scheme is stopped when the norm of the residual at the h-step, restricted to the entries related to T_j satisfies the criterion:

$$\left\| \left(\mathbf{r}_{\mathbf{y_i}}^h\right)_{|T_j} \right\| \leq \frac{c}{\sqrt{K\left(A_i^0\left(T_j, T_j\right)\right)}} \left\|\mathbf{g}_i\right\|, \quad i = 1, 2, 3, \quad 0 < c < 1$$

where $\mathbf{y}_1 = \mathbf{s}, \mathbf{y}_2 = \mathbf{z}$ and $\mathbf{y}_3 = \mathbf{u}$. From the practical point of view, since $\left\| \left(\mathbf{r}_{\mathbf{y}_i}^h\right)_{|T_j} \right\| \leq \left\|\mathbf{r}_{\mathbf{y}_i}^h\right\|$ and $K\left(A_i^0\left(T_j, T_j\right)\right) \leq K\left(A_i^0\right)$, in the standard stopping criterion of PCG (1) one must replace the norm of the gradient of the functional with respect to the block variable $\mathbf{y}_i$ at $\mathbf{x}_{S_j}^0$ with the one of corresponding sub-vector $\|\mathbf{g}_i\|$ related to T_j. Furthermore, at the first step of BCDA, a sufficient decrease of the objective function restricted to variables $(\mathbf{y}_i)_{T_j}$ is assured if the step-lenghts α_i^0 satisfy (2.1.2.5):

$$\alpha_i^0 \leq 2\left(1 - \rho_i\right) \frac{-g_i^T \left(\mathbf{d}_i^0\right)_{|T_j}}{\left(\mathbf{d}_i^0\right)_{|T_j}^T A_i^0\left(T_j, T_j\right)\left(\mathbf{d}_i^0\right)_{|T_j}}$$

Thus, since $\left(\mathbf{d}_i^0\right)^T A_i^0 \mathbf{d}_i^0 \geq \left(\mathbf{d}_i^0\right)_{|T_j}^T A_i^0\left(T_j, T_j\right)\left(\mathbf{d}_i^0\right)_{|T_j}$, we set $\alpha_i^0 = \gamma_i \frac{-g_i^T \left(\mathbf{d}_i^0\right)_{|T_j}}{\mathbf{d}_i^0 A_i^0 \mathbf{d}_i^0}$.

Using the same argument as for the proofs of theorems $2,3$ we have

$$\left\|\nabla_{\mathbf{x}_{T_j}} F_\epsilon(\mathbf{x}^\ell)\right\| \leq \frac{(M+\frac{1}{\sqrt{C_2}})}{\sqrt{C_1}}\sqrt{f_{\mathbf{x}_{S_j}}(\mathbf{x}^0_{T_j};\mathbf{x}^0_{B_j}) - f_{\mathbf{x}_{S_j}}(\mathbf{x}^1_{T_j};\mathbf{x}^0_{B_j})} \tag{2.9}$$

In the next iterations of BCDA, the control at the Step $2.2(b)$ assures at each step a decrease of the objective function with a maximum value to $C_1\left\|\mathbf{x}^k_{T_j}-\mathbf{x}^{k+1}_{T_j}\right\|^2$. If the following condition

$$f_{S_j}\left(\mathbf{x}^k_{T_j};\mathbf{x}^0_{B_j}\right) - f_{S_j}\left(\mathbf{x}^{k+1}_{T_j};\mathbf{x}^0_{B_j}\right) \geq C_1\left\|\mathbf{x}^k_{T_j}-\mathbf{x}^{k+1}_{T_j}\right\|^2$$

is satisfied, the update of the iteration is performed and a new iteration starts. Otherwise, we put $\overline{\mathbf{x}_j} = \mathbf{x}^k_{T_j}$ and the inner solver is stopped. Indeed, 2.1.2.5 assumes the role of an inner stopping criterion. As consequence, at any outer ℓ-iteration we have that

$F_\epsilon(\mathbf{x}^{\ell+1}) \leq F_\epsilon(\mathbf{m}^j) \leq F_\epsilon(\mathbf{x}^\ell))\quad j=1,\ldots,p$

and, from 2.9 2.1.2.5 and the connection rule, the inequality 2.1.2.5 remains satisfied with $C_3 \leq \frac{C_1}{\left(M+\frac{1}{\sqrt{c_2}}\right)^2}$:

$$C_3\left\|\nabla_{\mathbf{x}_{T_j}} F_\epsilon\left(\mathbf{x}^\ell\right)\right\|^2 \leq \left(F_\epsilon\left(\mathbf{x}^\ell\right) - F_\epsilon\left(\mathbf{m}^j\right)\right) \leq \left(F_\epsilon\left(\mathbf{x}^\ell\right) - F_\epsilon\left(\mathbf{x}^{\ell+1}\right)\right)$$

Thus, in view of Theorem 3.5 in [209], we can obtain the following convergence result:

Theorem 6. *A limit point of the generated sequence $\{\mathbf{x}^\ell\}$ is stationary point of F_ϵ. Moreover, since $\mathcal{L}$ is compact, there exists at least a limit point of $\{\mathbf{x}^\ell\}$ and we have $\nabla F_\epsilon(\mathbf{x}^\ell)\to 0$ as $\ell\to\infty$.*

Before proving that the whole sequence $\{\mathbf{x}^\ell\}$ converges to some critical point of F_ϵ in $\mathcal{L}$ we recall a property of functions having the Kurdyka Lojasiewicz (KL) property, proved in [38, 236]

Theorem 7. *Let $\Omega\in\mathbb{R}^n$ be a compact set and let $f:\mathbb{R}^n\to(-\infty,\infty)$ be a proper and lower semicontinuous function. Assume that f is constant on Ω and satisfies the KL property at each point of Ω. Then, there exists $\sigma>0$, $\eta>0$ and a continuous and concave function $\phi:[0,\eta)\to[0,\infty)$, which is continuous on*

$(0, \eta)$ *and satisfies* $\phi(0) = 0, \phi' > 0$ *on* $(0, \eta)$*, such that*

$$\phi(f(\mathbf{x}) - f(\overline{\mathbf{x}}))dist(0, \partial f(x)) \geq 1$$

for every $\overline{\mathbf{x}} \in \Omega$ *and every* $\mathbf{x}$ *such that* $dist(\mathbf{x}, \Omega) < \sigma$ *and* $f(\overline{\mathbf{x}}) < f(\mathbf{x}) < f(\overline{\mathbf{x}}) + \eta$.

By following similar arguments to those of Theorem 1 in [236], we can prove the convergence of the whole sequence $\{\mathbf{x}^\ell\}$ generated by algorithm 2.

Theorem 8. *Given a starting point* $\mathbf{x}^0$*, the sequence* $\{\mathbf{x}^\ell\}$ *generated by generated by algorithm 2 has finite length, i.e.* $\sum_{\ell=0}^{\infty} \left\|\mathbf{x}^{\ell+1} - \mathbf{x}^\ell\right\| < \infty$ *and, thus, it converges to some critical point of* F_ϵ *in* $\mathcal{L}_{F_\epsilon^0}$.

Following [126] we can obtain the following results about the convergence rate of algorithm 2.

Theorem 9. *Let* $\Phi : (0, \eta) \to [0, \infty)$ *be any primitive of* $-(\phi')^2$.

i. If $\lim_{\ell \to \infty} \Phi$ *is finite, the algorithm converges in a finite number of steps.*

ii. If $\lim_{\ell \to \infty} \Phi = \infty$ *there exists* $\overline{\ell}$ *such that*

- $F_\epsilon(\mathbf{x}^\ell) - F_\epsilon(\overline{\mathbf{x}}) = \mathcal{O}(\Phi^{-1}((\ell + 1 - \overline{\ell}))\frac{C_1}{Kp^2C_4^2}$
- $\left\|\mathbf{x}^\ell - \overline{\mathbf{x}}\right\| = \mathcal{O}(\phi \circ \Phi^{-1}((\ell + 1 - \overline{\ell}))\frac{C_1}{Kp^2C_4^2}$

The proof follows from the same argument of Theorem 3.5 Eq. (9) in [126].

The theoretical results about the convergence and the rate of convergence of algorithm 2 do not depend on the size ν of the frame of each tile T_j. Nevertheless, practical experience shows that a selection of $\nu > 0$ decreases the number of external iterations. When $\nu > 0$ the iterations are few, less than ten, also for huge images. Indeed, in the connection rule, the selection $\nu > 0$ facilitates the choice $\mathbf{x}^{\ell+1} = \tilde{\mathbf{m}}$ rather than $\mathbf{x}^{\ell+1} = \mathbf{m}^j$, for a suitable j, fully exploiting the parallel processing of each tile.

As we can see from algorithm 1, the main computational bottleneck of BCDA algorithm is the inexact minimization step of the functional restricted to one of the blocks $(\mathbf{u}, \mathbf{s}, \mathbf{z})$: few iterations of a PCG algorithm are required for the evaluation of the descent direction. For taking advantage of all the cores in a commodity **CPU**, parallel linear algebra libraries can be then adopted to

speed up the segmentation problem. To this extent we adopted **Thrust** library (`https://thrust.github.io/`): that offers the possibility to target different architectures by selecting, at compile-time, the parallelization backend. Besides **GPU** support, a multicore **CPU** approach is offered based on **OpenMP**. The library takes care of the parallelization of linear algebra routines: in this BCDA implementation we then used Thrust provided implementation of vector norms and vectors updates (axpy). Concerning the evaluation of the elements of matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$, custom code has been developed and parallelized through *OpenMP parallel for* directive. Diagonal preconditioner and linear Conjugate Gradient are already offered by the library that offers the possibility to introduce custom preconditioners. This chance is exploited to develop a block tridiagonal symmetric preconditioner class: factorization step can be easily parallelized, since we are facing a block-diagonal structure. Matrix-vector product routines involved banded matrices: a tabular structure is used to memorize diagonals in compact vectors. Implementation relies on **Cusp** library (`https://cusplibrary.github.io/`), that provides dedicated *dia_matrix* class. It is useful to note that underlying implementation resides on aforementioned Thrust blas routines. This first parallelization approach allows one to implement a version of BCDA that contains parallel matrix vector product subroutines, plus a number of $blas - 1$ subroutines for norms and dot products. When varying the number of threads, memory-bound nature of this problem inhibits a satisfactory decrease of the overall computational time: parallelization speed up is low due to poor local data reuse. It is worth noting that, even if each subroutine is able to split the computational task among more than one core, there is no data reuse: each matrix-vector evaluation involves a complete scan of $(\mathbf{u}, \mathbf{s}, \mathbf{z})$ vectors, no temporal locality on data access is exploited. As a consequence, when large images are considered, no speed up is achieved. The second parallelization approach aims at increasing data locality by partitioning and considering independent subproblems. Following this strategy, we are facing a decrease of data dimensionality and variables are more likely to fit in the hardware cache, thus leveraging the impact of extensive memory access. Concerning the algorithm 2 the intrinsic features of steps 2 and 3 are exploited in two different ways. The tilling technique previously described is exploited in order to generate at step 2 a number of **independent tasks** that can be concurrently solved; concerning step 3 $openMPparallelfor$ directive is used when evaluating the objective function. Due to iterative nature

of the inner BCDA solver, different running times are expected for the solution of subproblems: to overcome this inconvenience we adopted manager/worker pattern [17] that ensures runtime distribution of independent tasks among POSIX threads. A number of computational threads (workers) is initialized and put on wait on a shared task queue, while a monitor thread (master) is responsible to extract for each subproblem j, initial data $\mathbf{w}_j^0$ from current solution $\mathbf{x}^\ell$ and collects subproblems computed solutions. Mutex-protected queues collect both task input and output results, as consequence two different queues are implemented:

- a job queue: a single manager is the producer of the queue elements, while all workers are consumers;
- a results queue: in this case each worker fills the queue with results of assigned subproblems, while the manager is responsible to insert them in the overall segmentation variables $(\mathbf{u}, \mathbf{s}, \mathbf{z})$.

Both cases can be handled by the same implementation that provides:

- a thread-safe interface for insert/remove operations;
- a signaling mechanism for the communication of available resources;

An additional C++ class that stores resources in a private $std :: queue < T >$ variable, while exposes only two methods push and pop for resource insertion and removal. This implementation can be used in conjunction with POSIX thread [55] since additional members are present:

- a mutex variable of type $pthread_mutex_t$ used as safeguard for the shared resource;
- a condition variable of type $pthread_cond_t$, associated to previously mentioned mutex, for signaling procedures.

Such implementation choice allows for a mutually exclusive access to internal queue in multi-threading environment. Moreover, through the adoption of a condition variable, producer threads can communicate information about the state of shared data: for example to signal that a queue is no longer empty. An exhaustive description of this approach can be found on Chapter 3 of [55]. In

order to provide a reliable queue implementation even in the presence of exceptions, RAII (Resource Acquisition Is Initialization) [301] programming idiom is adopted when locking/unlocking operations are executed on a mutex. Job queue is used to communicate both commands and data from master to workers: in this implementation, only two basic job types are used. A first job type contains a complete description of one of the tasks (references to subproblem local data, objective function parameters and algorithm parameters) generated in Step 2 of Algorithm 2. A second type of job is used by master thread in order to ensure the clean termination of workers threads. Each worker thread is structured as a while loop: as long as the thread can pick a sub problem description, it solves it and puts the outputs on results queue; when a termination job is picked, the thread exits. Finally, as regard Step 3, **OpenMP** compiler directive *omp parallel for* is used for evaluation of $F_\epsilon(\widetilde{\mathbf{m}})$. While this step is performed, worker threads are waiting on a POSIX condition variable, without requiring **CPU** time.

2.2 Variational Methods on Graphs

2.2.1 Discrete calculus

Discrete calculus [152, 162] has been used in recent years to produce a combinatorial reformulation of continuous problems onto a graph in such a manner the solution behaves analogously to the continuous formulation [106, 150].

Discrete calculus has to be differentiated from *finite elements discretization*. Such discretization aims to produce approximate solutions to problems defined in the continuum. The approximation becomes better as the discretization becomes finer and finer. In contrast, discrete calculus does not refer to the continuum.

We define some terminologies that will be used in the following sections.

A graph consists of a pair $G = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$. Let n represent the number of nodes, i.e. $n = |V|$, and m the number of edges of G, i.e. $m = |E|$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. We deal with weighted graphs that include weights on both edges and nodes. An edge weight is a value assigned to each edge e_{ij}, and is denoted by w_{ij}. We assume $w_{ij} \in \mathbb{R}_+^*$ and use w to denote the vector of $\mathbb{R}^m$ containing the w_{ij} for every edge e_{ij} of G. In addition to edge weights, we may also assign weights to nodes. The weights of a node v_i is denoted by g_i. We assume $g_i \in \mathbb{R}_+^*$ and

use g to denote the vector of $\mathbb{R}^n$ containing the g_i for every node v_i of G. Usually vertices represent image elements, which might represent pixels or region in the Euclidean space and weights measure a degree of similarity between to vertices connected by an edge.

The incidence matrix of graph is a key operator for defining combinatorial formulations of variational problems. Specifically, the incidence matrix $A \in \mathbb{R}^{n \times m}$ is known to define the discrete calculus analogue of the gradient, while A^T is known to define the discrete calculus analogue of the divergence (see [152] and the references therein). The incidence matrix maps functions on nodes (a scalar field) to functions on edges (a vector field) and may be defined as

$$A_{e_{ij}v_k} = \begin{cases} -1 & \text{if } i = k, \\ +1 & \text{if } j = k, \\ 0 & otherwise. \end{cases}$$

for every vertex v_k and edge e_{ij}. As explained in [152], several important theorems such as the fundamental theorem of calculus or the Helmholtz decomposition hold in this discrete framework.

2.2.2 Regularization methods

Numerous problems in signal processing and computer vision involve the optimization of an energy function composed of a regularization term and a data fidelity term. For example, image denoising problems may be solved by considering that the optimal recovered image is somehow regularized, that is to say that variations (the local differences) of intensity are limited. For image segmentation problems, the sum of variations between neighboring labels may be penalized. In stereovision variational approaches, estimation of depth maps also assumes a local consistency for the depth and limit the sum of variations in order to obtain a piecewise smooth estimation.

In all these cases, the problem may be expressed as follows. A labelling x is estimated, x being the solution to the minimization of its total variations added to a term enforcing data fidelity.

$$\min_x \int_\Omega \underbrace{||\nabla(z)||\, dz}_{\textbf{Regularization}} + \underbrace{D(x)}_{\textbf{Data fidelity}} \tag{2.10}$$

A simple example of data fidelity term in image denoising context could be

$$D(x) = \int_{\Omega} (x(z) - f(z))^2 \, dz, \tag{2.11}$$

appropriate in the case where f is an image corrupted with Gaussian noise. We now review shortly two classical and efficient regularization models.

2.2.2.1 Total variation

The Total Variation problem (TV) was introduced originally in computer vision by Shulman and Herve [293] as a regularizing criterion for optical flow computation and later Rudin, Osher and Fatemi [270] as a regularizing criterion for image denoising. It has been shown to be quite efficient for smoothing images without affecting contours so much. Moreover, a major advantage of TV is that it is a convex problem, making it possible to find a global minimizer. When applied to a 2D discrete image $(x_{i,j}), 1 \leq i, j \leq N$, the total variation minimization problem becomes

$$\min_{x} \sum_{1 \leq i,j \leq N} ((x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2)^{\frac{1}{2}} + D(x) \tag{2.12}$$

using a finite-difference discretization popularized in [67]. The Total variation problem has been expressed in weighted and non-local graphs in [39, 139], leading to better penalization of the gradient.

A number of convex optimization techniques suitable for solving this problem has been employed for several decades. The most recent and efficient approaches are compared in [68].

Among the most efficient methods, one can cite Nesterov's algorithm [232], Split-Bregman / Douglas-Rachford methods [139, 143], and Chambolle-Pock's Primal-dual algorithm [71]. Most methods minimizing TV focus on image filtering as application, and even if those methods are remarkably fast in denoising applications, in segmentation problems require more iterations for those algorithms to converge.

2.2.2.2 Max Flow - Min Cut

The max-flow/min-cut problem on a graph is a classical problem in graph theory, for which the earliest solution algorithm goes back to Ford and Fulkerson [122]. Initial methods for global optimization of the boundary length of a region formulated the energy on a graph and relied on max-flow/min-cut methods for solutions [122]. In the context of image segmentation, the seeded max-flow/min-cut model is also known as "graph cuts". Graph cuts algorithms provide a mechanism for discrete optimization of an energy functional [185], which have been used in a variety of applications such as image segmentation, stereo, image stitching and texture synthesis. More specifically, the labelling x produced by graph cuts algorithms is an optimal (not necessarily unique) solution to

$$\min_{x \in 0,1^n} \sum_{e_{i,j} \in E} w_{ij} |x_i - x_j| + D(x) \tag{2.13}$$

This energy may be optimized using the non-polynomial, but experimentally fast methods of [44]. As explained in [186], more general - submodular - energies than 2.13 may be optimized using graph cuts. Consequently, the graph cuts technique has been adapted to multilabel problems using various graph constructions. One notable construction is that of Ishikawa [168] presenting the multilabel problem as a segmentation problem.

2.2.2.3 Combinatorial Dirichlet Problem

The Laplace equation arises in many physical situations: heat propagation, fluid dynamic and electronics among others. The Dirichlet problem aims at finding a solution satisfying the Laplace equation subject to some boundary constraints.

In computer vision and image processing problems, solving the Dirichlet problem provides effective solution to labelling problems, given some markers. In this context, the combinatorial Dirichlet problem is written

$$\min_{x \in \mathbb{R}^n} \sum_{e_{i,j} \in E} w_{i,j} (x_i - x_j)^2 + D(x) \tag{2.14}$$

In image processing, this problem has been applied for example to interpolation [153], impainting [62], image filtering [33], and seeded image segmentation [148]. The famous "Random Walker" algorithm [148] is an adaptation of Dirichlet

problem to image segmentation.

2.2.2.4 Free Discontinuity Problems on Graphs

As seen in the previous section, the Mumford-Shah functional (MSF) formulates the problem of finding piecewise smooth reconstructions of functions (e.g., images) as an optimization problem [229]. Optimizing the MSF involves determining both a function and a contour across which function smoothness is not penalized. Unfortunately, since smoothness of the reconstruction is not enforced across the contour and since the contour is variable in the optimization, the functional is not easily minimized using classical calculus of variations.

Given a fixed contour it is possible to solve for the optimal reconstruction function by solving a straightforward elliptic PDE with Neumann boundary conditions. Additionally, given a fixed piecewise smooth reconstruction function, it is possible to determine, at each point on the contour, the direction that the contour would move to decrease the functional as quickly as possible. Thus, most methods for solving the MSF involve alternating optimization of the reconstruction function and the contour [72], [73], [308]. The results of performing this type of optimization are well known and achieve satisfactory results that are used for different imaging applications [308]. Unfortunately, this optimization of the MSF using contour evolution techniques (typically implemented with level sets) is slow primarily due to the small steps taken by the contour at each iteration. This slowness is exacerbated by the fact that a small perturbation of the contour can have a relatively large effect on the optimal reconstruction function. Additionally, these traditional methods often require many implementation choices (e.g. implementation parameters) and these choices may produce differences in the final result. Practical energy minimization problems formulated on a finite set of variables can be solved efficiently using combinatorial algorithms [154], [186], [294]. Furthermore, because of the well-established equivalence between the standard operators of multidimensional calculus and certain combinatorial operators, it is possible to rewrite many PDEs formulated in $\mathbb{R}^N$ equivalently on a graph. Reformulating the conventional, continuous, PDE on a graph permits straightforward application of the arsenal of combinatorial optimization techniques to efficiently solve these variational problems. An alternate view of our approach is to consider rewriting the continuous energy functional in terms of the precise discrete

operations that would be performed on a computer to evaluate the energy of a particular solution. By writing this energy in discrete terms, we can design our optimization method to optimize the energy value that would actually be measured by the computer. In this section we reformulate the MSF on a graph so that we may apply a combinatorial optimization to reduce the difficulties of speed and local minima associated with the small contour improvements obtained via traditional contour evolution. An added benefit of reformulating an energy in a combinatorial setting is that such a generic formulation may be applied without modification to higher dimensional data or general data analysis problems, such as point clustering, mesh smoothing/segmentation or space-variant vision. Similar approaches have been employed successfully in the minimization of total variation methods [106]. We now describe the combinatorial analogue of the piecewise smooth Mumford-Shah model. We recall that the degree of a vertex in a graph G is $d_i = \sum w(e_i j)$ for all edges $e_i j$ incident on v_i. An image may be associated with a graph by identifying each pixel with a node and defining an edge set to represent the local neighborhood of the pixels (e.g. a 4-connected lattice). Since its inception, there have have been several related notions of what constitutes the Mumford-Shah functional. In this case we follow the level set literature to consider the piecewise smooth model formulated as:

$$E(f, g, R) = \alpha\left(\int_R (f-p)^2 + \int_{\Omega\setminus R} (g-p)^2\right) + \mu\left(\int_R ||\nabla f||^2 + \int_{\Omega\setminus R} ||\nabla g||^2\right) + \nu\Gamma(R).$$

where Ω represents the image domain, f is the smoothed foreground function, g is the smoothed background function, R is the region of the image comprising the foreground, p is the pixel intensity, $\Gamma(R)$ is a function returning the length of the contour of region R, and α, μ, ν are free parameters. To simplify the parameter space, we assume that all three free parameters are strictly positive and divided by the value of μ. Thus, we will omit the inclusion of μ in the remaining part. Similar models were considered by Blake and Zisserman, who referred to the energy as the "weak membrane model" [35] and by the influential paper of Geman and Geman [136].

Formulation of 2.2.2.4 on a graph requires the use of combinatorial analogues of the continuous vector calculus operators. Although combinatorial representations of differential operations are fairly well established, the challenge in the

graph reformulation of any particular energy (or PDE) is to associate variables in the continuous formulation with representative combinatorial structures (pixels, edges, cycles, etc) and, as in the continuous case, to produce a useful representation of a "contour". Specifically, each integral may be considered as a paring between a chain (domain of integration) and a cochain (function to be integrated). Associating each pixel in our image with a node in the graph, the integration over a collection of pixels (in set $S_R \subseteq V$) may be represented by the $N \times 1$ chain vector r, where

$$r_i = \begin{cases} 1 & \text{if } v_i \in S_R, \\ 0 & \text{otherwise.} \end{cases}$$

The other two variables in E are cochains taking real values, i.e., $f_i \in \mathbb{R}, g_i \in \mathbb{R}$. Note also that the image I is treated as a vectorized, real-valued cochain existing on the nodes(pixels). Both chains and cochains will be treated as column vectors.

The first (data) term in 2.2.2.4 concerns quantities associated with pixels (i.e., intensities). We chose above to associate nodes with pixels, so p,f, and g must represent a 0-cochain (a function mapping nodes to real numbers). This matches the continuous conception of these quantities as scalar fields. Since the data term in 2.2.2.4 integrates over a set of the domain for which p, f and g are defined, r must represent a 0-chain indicating a region of the domain. Thus, the analogue of the first term on a graph is:

$$E_1(f, g, r) = r^T(f - p)^2 + (1 - r)^T(g - p)^2$$

In order to formulate the second term one should recall that the combinatorial analogue of the gradient operator is given by the node-edge incidence matrix, A. Consequently, we may write the gradient of f as the product Af. However, since gradients are $vectorfunctions$ (corresponding to cochain on edges in the combinatorial setting) and the integral in the second term is performed over a scalar (i.e., the norm of the gradient at each point), we have to transfer the gradient cochain associated with edges back to a scalar cochain associated with nodes. Such an operator may be represented by the absolute value of the incidence matrix, although each edge is now double counted, requiring a normalizing factor

of one-half. Specifically, the second term may be formulated as

$$E_2(f,g,r) = \frac{1}{2}\Big(r^T|A|^T(Af)^2 + (1-r)^T|A|^T(Ag)^2\Big). \tag{2.15}$$

Finally, the contour length term may be formulated on a graph by counting the edges spanning from R to $\overline{R}$. Such a measure may be represented in matrix form as:

$$E_3(f,g,r) = 1^T|Ar|. \tag{2.16}$$

If our graph is a standard 4-connected lattice, then 2.16 produces the ℓ_1 measure of the region R. If we view the graph as embedded in $\mathbb{R}^{\mathbb{N}}$ and wish to measure the Euclidean contour length, it was shown [43] that a suitably weighted graph and corresponding incidence matrix could instead be used in 2.16. However, since this construction was designed to produce a Euclidean contour length, we use this construction only in term E_3. For purposes of generality and clarity here, we will continue to use the same A in all terms. All three terms may now be put back together to define the combinatorial analogue of the piecewise smooth Mumford-Shah model, i.e.,

$$E(f,g,R) = \alpha\Big(r^T(f-p)^2 + (1-r)^T(g-p)^2\Big) + \frac{1}{2}\Big(r^T|A|^T(Af)^2 + (1-r)^T|A|^T(Ag)^2\Big) + \nu 1^T|Ar|$$

2.2.3 Combinatorial Optimization Techniques

A good overview of combinatorial optimization techniques is presented in [243]. We described those are used in the segmentation methods described in the next chapter.

2.2.3.1 Maximum Flow/Minimum Cut Algorithms

A transport graph G is a graph with exactly one source s, with no entering edge, and one sink t, with no exiting edge. A flow in a transport graph is a function associating a value to every edge according to some rules. The value of the flow in each edge must be less than the capacity (weight) of the edge, and the sum of the flow entering in each node - except s and t - must be equal to the sum of flow

exiting the node. The max-flow problem consists of maximizing the amount of flow sent from the source toward the sink. This process creates a bottleneck for some edges. If the flow of an edge is equal to its capacity, we say that the edge is saturated.

A cut is a partition of the nodes into two sets S and T, such that s is in S and t is in T. The capacity of a cut is the sum of the weights of all the edges crossing the cut, from the region S to the region T. Ford and Fulkerson [122] proved that maximum flow is equals to the capacity of a minimal cut.

Most max-flow algorithms can be sorted in two categories, augmenting path [122] and push-relabel methods [142]. Min-cut algorithms without computing a maximum flow exist, a well known one is Stoer-Wagner algorithm [297]. Its complexity of $O(nm + n^2 log(m))$ is one of the most effective for general purpose min-cut/max-flow methods, but in practice the runtime on image data is not the fastest. The regular lattice structure of images was exploited in the dedicated max-flow algorithm of [44], resulting in faster runtimes.

2.2.3.2 Shortest Path Methods

The shortest path problem consists of finding, from two nodes of a graph, a path of minimal cost (such that the sum of weights of component edges is minimized) in the graph. When the graph does not contain negative weights, Dijkstra's algorithm [96] is very efficient. When implemented using Fibonacci heaps, its complexity is $O(m + nlog(n))$. First introduced as a by-product of reinitializing level set energies, the fast marching algorithm [288] has been used for shortest path computation [79] over sampled continuous domains.

Shortest path methods are popular in image processing, particularly for thin object segmentation. The "Intelligent scissors" [227] consists, for example, of computing shortest paths between several points placed on the contour of objects to be extracted. Improvements have been proposed to limit the user interaction, and to define better weighting strategies [254]. Other strategies for image segmentation using shortest path based algorithms include the construction of shortest path forests. Given foreground and background seeds, each pixel is assigned to the foreground label if there is a shorter path from that pixel to a foreground seed than to any background seed. This approach was recently popularized by Bai and Sapiro [21].

2.2.3.3 Maximum Spanning Trees and Forests

A spanning tree of a graph is a tree (a connected graph without cycles) connecting all the nodes of the graph. The weight of a spanning tree is defined by the sum of the edge weights composing the tree. There exist several greedy algorithms minimizing or maximizing this weight, called respectively minimum or maximum spanning trees algorithms. The Prim [258] and Kruskal [190] algorithms are the most commonly used. When using a union-find data structure [305] for cycles detection, the complexity of maximum spanning tree algorithms is quasi-linear [74].

In clustering applications, a set of several maximum/minimum spanning trees may be computed, where different trees correspond to the different defined labels. The resulting set of trees is called a Maximum (or resp. Minimum) Spanning Forest. The segmentation is given by the Maximum Spanning Forest cut, defined by the set of edges that separates different trees. The first appearance of such forest in image processing dates from 1986 with the work of Morris [226]. It was later introduced by Meyer in a morphological context in [218]. Different criteria for regions merging appear in the literature, as for example in the widely used algorithm of Felzenszwalb [117]. If the markers are located on the minima of the weight function, the cut obtained by minimum spanning forest computation was shown by Cousty to be a watershed cut [82]. These greedy procedures may be used for optimizing meaningful and useful energies.

2.3 Conclusions

One of the scopes of this work is to propose a novel method to segment thin structures such as cracks in images. One of the best known model for image segmentation was introduced by Mumford and Shah ([229]) and belongs to variational methods. This one is successful in segmenting object edges that are related to discontinuities in grey level intensities, but has some drawbacks: it is unable to reconstruct crease discontinuities and yields the over-segmentation of steep gradients (the so-called ramp effect). To overcome these defects, Blake and Zisserman [35] introduced a second order functional, but its minimization is extremely difficult. However, Γ-convergence proposed by Ambrosio and Tortorelli has shown to be fundamental to solve the problem of numerically computing a minimizer. The

main result of this chapter is a parallel approach for this minimization technique that is of remarkable importance when large-size images are considered. Results on a crack dataset are showed in Section 5.2.2 in Chapter 5. Besides, we showed how variational methods can be easily extended to graphs in order to process different kinds of data.

Chapter 3

Segmentation on Graphs

3.1 Introduction

One of the most classical and fundamental problems in computer vision is image segmentation. It refers to partitioning an image into several disjoint subsets such that each subset corresponds to a meaningful part of the image. As an integral step of many computer vision problems, the quality of segmentation output largely influences the performance of the whole vision system. A rich amount of literature on image segmentation has been published over the past decades. Some of them have achieved an extraordinary success and become popular in a wide range of applications, such as medical image processing [151, 170], object tracking [286], recognition [296, 273], image reconstruction [116, 36] and so on. Since the very beginning, image segmentation has been closely related to perceptual grouping or data clustering. Such a relationship was clearly pointed out by Werterheimer's gestalt theory [327] in 1938. In this theory, a set of grouping laws such as similarity, proximity and good continuation are identified to explain the particular way by which the human perceptual system groups tokens together. The gestalt theory has inspired many approaches to segmentation and it is hoped that a good segmentation can capture perceptually important clusters which reflect local and/or global properties of the image. Early edge detection methods such as the Robert edge detector, the Sobel edge detector [146] and the Canny edge detector [56] are based on the abrupt changes in image intensity or color. Due to the distinguishable features of the objects and the background, a large number of thresholding based methods [328, 329, 266, 330] have been proposed

to separate the objects from the background. In the partial differential equations (PDE) based methods [288, 177, 241, 240, 72, 239, 84], the segmentation of a given image is calculated by evolving parametric curves in the continuous space such that an energy functional is minimized for a desirable segmentation. Region splitting and merging is another popular category of segmentation methods, where the segmentation is performed in an iterative manner until some uniformity criteria [234, 103] are satisfied. Among the previous image segmentation techniques, many successful ones benefit from mapping the image elements onto a graph. The segmentation problem is then solved in a spatially discrete space by the efficient tools from graph theory. One of the advantages of formulating the segmentation on a graph is that might require no discretization by virtue of purely combinatorial operators and thus incur no discretization errors.

With a history dating back to 1960s, the earliest graph theoretic methods stress the importance of the gestalt principles of similarity or proximity in capturing perceptual clusters. The graph is then partitioned according to these criteria such that each partition is considered as an object segment in the image. In these methods, fixed thresholds and local measures are usually used for computing the segmentation results, while global properties of segmentation are hard to guarantee. The introduction of graph as general approach to segmentation with a global cost function was brought by Wu et al. [335] in 1990s. From then on, much research attention was moved to the study of optimization techniques on the graph. It is known that one of the difficulties in image segmentation is its ill-posed nature. Since there are multiple interpretations of the image content, it might be difficult to find a single correct answer for segmenting a given image. This suggests that image segmentation should incorporate the mid- and high-level knowledge in order to accurately extract objects of interest. In the late 1990s, a prominent graph technique emerged in the use of a combination of model-specific cues and contextual information. An influential representation is the s/t graph cut algorithm [46]. Its technical framework is closely related to some variational methods [288, 177, 241, 240, 72, 239, 84] in terms of a discrete manner. Up to now, s/t graph cut and its variants have been extended for solving many computer vision problems and eventually acting as an optimization tool in these areas.

In the next section a systematic survey of graph theoretic techniques is provided. They are grouped into five categories. (1) *Minimal spanning tree-based*

methods: the clustering or grouping of image pixels is performed on the minimal spanning tree. The connection of graph vertices satisfies the minimal sum on the defined edge weights and the partition of a graph is achieved by removing edges to form different subgraphs. (2) *Graph cut with cost functions*: graph cut is a natural description of image segmentation. Using different cut criteria, the global functions for partitioning the graph will be different. Usually, by optimizing these functions, we can get the desirable segmentation. (3) *Graph cut on Markov random field models*: the goal is to combine the high-level interactive information with the regularization of the smoothness in the graph cut function. Under the MAP-MRF framework the optimization of the function is obtained by the classical min-cut/max-flow algorithm or its nearly optimal variants. (4) *The shortest path-based methods*: the object boundary is defined on a set of shortest paths between pairs of graph vertices. These methods require user interactions to guide the segmentation. (4) *Other methods*: several efficient graph theoretic methods that do not belong to any of the above categories are described, such as random walker [147] and dominant set-based method [244].

3.2 Classical Methods

In this section we review the most representative graph segmentation methods. For each class, the formulation of the problem is provided and an overview of their implementation is presented. Their advantages and disadvantages are discussed as well. Although the methods are classified into five categories, some of them are often used in conjunction with one another.

First, we recall some useful definitions (see also subsection 2.2.1). For image segmentation an image is partitioned into mutually exclusive components such that each component A is a connected graph $G' = (V', E')$, where $V' \subset V, E' \subset E$ and E' contains only edges built from nodes of V'. In other words, non-empty sets $A_1, \ldots, A_k$ form a partition of the graph G if $A_i \cap A_j = \emptyset$ and $A_i \cup \ldots \cup A_j = G$. The well-accepted segmentation criteria [327] require that image elements in each component should have uniform and homogeneous properties in the form of brightness, color, texture etc. and elements in different components should be dissimilar.

In graph theoretic definition the degree of dissimilarity between two compo-

nents can be computed in the form of a graph cut. A cut is related to a set of edges by which the graph G will be partitioned into two disjoint sets A and B. As a consequence the segmentation of an image can interpreted in form of graph cut, where the cut value is usually defined as:

$$cut(A, B) = \sum_{u \in A}^{v \in B} w(u, v) \tag{3.1}$$

where u and v refer to the vertices in two different components. In image segmentation noise and other ambiguities bring uncertainties into the understanding of image content. The exact solution to image segmentation is hard to obtain. Therefore it is more appropriate to solve this problem with optimization methods. The optimization-based approach formulates the problem as the minimization of some established criterion, whereas one can find an exact or approximate solution to the original uncertain visual problem. In this case, the optimal bi-partitioning of a graph can be taken as the one which minimizes the cut value in Eq. 3.3.1.1.

In a large amount of literature image segmentation is also formulated as a labelling problem, where a set of labels L is assigned to a set of sites in S. In two-class segmentation, for example, the problem can be described as assigning a label f_i from the set $L = \{object, background\}$ to site $i \in S$ where the elements in S are the image pixels or regions. Labelling can be performed separately from image partitioning while they achieve the same effect on image segmentation. Several methods perform both partitioning and labelling simultaneously.

Methods in image segmentation can be categorized into **automatic** methods and **interactive** methods. Automatic segmentation is desirable in many cases for its convenience and generality. However, in many applications such as medical and biomedical imaging, objects of interest are often ill-defined so that even sophisticated automatic segmentation algorithms will fail. Interactive methods can improve the accuracy by incorporating prior knowledge from user; however, in some practical applications where a large number of images are needed to be handled, they can be laborious and time consuming. Note that automatic and interactive methods are often used together to improve the segmentation results. Some automatic segmentation methods may require interaction for setting initial parameters and some interactive methods may start with the results from automatic segmentation as an initial segmentation.

3.2.1 Minimal Spanning Tree based methods

The minimal spanning tree (MST) (also called shortest spanning tree) is an important concept in graph theory. A spanning tree T of a graph G is a tree such that $T = (V, E')$, where $E' \subseteq E$. A graph may have several different spanning trees and the MST is the one with the smallest weights. The algorithms for computing the MST can be found in [190, 97, 258]. For example, in Prim's algorithm the MST is constructed by iteratively adding the frontier edge of the smallest edge-weight. The algorithm is in a greedy style and runs in polynomial time. MST based segmentation methods are essentially related to the graph-based clustering. The general study of graph clustering can be dated back to 1970s or earlier. In graph-based clustering the data to be clustered are represented by an undirected adjacency graph. In order to represent the affinity, edges with certain weights are defined between two vertices if they are neighbors according to a given neighborhood system. Clustering is then achieved by removing edges of the graph to form mutually exclusive subgraphs. The clustering process usually emphasizes on the importance of the gestalt principles of similarity or proximity in the graph vertices.

The early MST based methods [342] perform image segmentation in an implicit way, which is based on the inherent relationship between the MST and the cluster structure. The intuition underlying this relationship is that the MST consists of edges with the minimal sum of weights among all spanning trees and, as result, it guarantees the connection of graph vertices which are most similar to each other (i.e., at the lowest cost of weights). This nature makes MST spans all the vertices and at the same time jump across the smaller gaps between different clusters. However, it is not enough to deal with situations when there is a large variation inside a cluster. The complex scenes in real world images often have perceptually meaningful clusters with non-uniform densities; therefore it is more desirable to consider both the difference across two clusters and the difference inside a cluster. The gestalt principles play an important role in guiding the MST based image segmentation; however, they lack a precise measurement on the definition in quantitative results.

Morris et al. [225] used MST to hierarchically partition images. Their method can obtain the segmentation in different scales based on the principle that the most similar pixels should be grouped together and dissimilar pixels should be

separated. By cutting the MST at the highest edge weights, partitions of a graph are formed with the maximal difference between neighbouring subgraphs. In some improved algorithms were also proposed based on MST, e.g. the recursive MST algorithm. In [225], some improved algorithms were also proposed based on MST, e.g. the recursive MST algorithm. In each iteration, the segmentation is formed by partioning one subgraph. Therefore, the algorithm can lead to a final segmentation with a given number of subgraphs. Apparently, the algorithm in this form is inefficient. Kwok et al. [191] proposed a fast recursive MST algorithm to speed up Morris et al.'s method.

An advanced work of MST based algorithm proposed in [117] makes use of both the differences across two subgraphs and the differences inside a subgraph. The segmentation is performed in conjunction with a region merging process and produces results that satisfy some global properties. The key of this algorithm is adaptive thresholding. In contrast to single linkage clustering which uses a constant K to set the threshold, here it is a variable and is defined on the size of clusters. This allows two components to be merged if the linkage between them is smaller than the maximal edge in either of the components' MST plus this threshold. The formal definition of the merging criterion is given as below:

$$|e_t| \leq min\left(Int(C_1) + \frac{K}{|C_1|}, Int(C_2) + \frac{K}{|C_2|}\right) \tag{3.2}$$

where K is a constant, $|C_1|$ and $|C_2|$ are the sizes of component C_1 and C_2, respectively. $Int(C)$ is the largest edge weight in the **MST** of C. $|e_t|$ is the edge with the smallest weight which connects C_1 and C_2. From Eq. 3.2 we can see that the algorithm is sensitive to edges in smooth areas and less sensitive to areas with high variability.

It is evident that in the context of edge-weighed graphs MST based algorithms explicitly define the structures of clusters. Pixels expressed by low level features such as intensity, colour or texture can be intuitively organized by these algorithms. However, the algorithms are strongly based on the assumption that labelling of pixels in the same segment is consistent. This is not the case when these pixels belong to different object classes. Therefore, this category of algorithms is often used as an initial processing for other high-level applications [165, 166]. MST often forms a segmentation by cutting it at the highest edge weights, so a further region can be obtained by making a further cut in the tree. This implies

a hierarchical segmentation in MST, which provides a mechanism for converting any over-segmentation into higher-level counterparts without loss of the cluster feature.

3.2.2 Graph Cut with Cost Functions

3.2.2.1 Minimal Cut Methods

Using graph cut for image segmentation was first proposed by Wu and Lethy [334] in 1990. Like MST, graph cut is also a notion explicitly defined on edge-weighted graphs. Graph cut based methods propose a general framework of optimally partitioning a graph globally. This brings the advantages that for different applications, different cost functions can be designed with a clear definition of segmented objects. Graph cut in equation 3.3.1.1 provides us with an opportunity for a clear and meaningful definition of graph partitioning: minimizing this cut makes vertices in different sets dissimilar. However, for a practical graph partition problem, it also requires vertices in the same set to be similar. These two requirements are studied by existing graph cut methods, which attempt to satisfy one or two of the requirements.

In their work [334] Wu and Leahy minimized a cost function formulated exactly in the form of equation 3.3.1.1, namely minimal cut. According to the Ford-Fulkerson theorem [123], the maximum flow between a pair of vertices equals to the value of the minimal $s/t - cut$, which could be solved efficiently. In [334] the authors also discussed a more general case where a k-partition of graph G is identified by using the Gomory-Hu algorithm [145], as an equivalent of finding the maximal flow between k-pairs of vertices.

3.2.2.2 Normalized Cut Methods

The minimal cut criterion gives a good illustration of gestalt principles, but it has a bias towards finding small components. In order to alleviate this problem, one should consider to explicitly require that each individual set is "reasonably large". Several studies have been done to address this problem, which lead to various normalized objective functions.

One well-known objective function to avoid this unnatural bias is proposed by Shi et al. [292] in terms of normalized cut (NCUT). The graph cut is measured

by the weights of $vol(\cdot)$, which is the total connection from vertices in a set (e.g. A) to all the vertices in the graph. Formally we have $vol(A) = \sum_{v_{i\in A}, v_j \in V} w(v_{i,j})$, where weight measures a certain image quantity (e.g. intensity, colour, etc.) between the two vertices connected by that edge. Then $Ncut$ cost function is defined as follows:

$$Ncut(A,B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)} =$$

$$\frac{\sum_{x_i>0, x_j<0} -w(v_i,v_j)x_i x_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0, x_j>0} -w(v_i,v_j)x_i x_j}{\sum_{x_i<0} d_i} \tag{3.3}$$

where x_i is the indicator variable, $x_i = 1$ if vertex v_i is in A and $x_i = -1$ otherwise. $d_i = \sum_j w(v_i, v_j)$ is the total connection from v_i to all the other vertices. Note that with this definition, the partitions containing small set of vertices will not have small $Ncut$ value and hence the minimal cut bias is circumvented. The minimization of equation 3.3 can be formulated into a generalized eigenvalue problem, which has been well-studied in the field of spectral graph theory. After a common matrix transformation, the $Ncut$ problem can be re-written into:

$$minNcut(A,B) = min_y \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})y}{\mathbf{y}^T\mathbf{D}y} \tag{3.4}$$

subject to $\mathbf{y}(i) \in \{1, -b\}, b = \frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$ and $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$, where $\mathbf{D}$ and $\mathbf{W}$ are the degree and the adjacency matrix of G, respectively. We call $\mathbf{L} = \mathbf{D} - \mathbf{W}$ the graph Laplacian of G. It can be seen that $-b$ represents the ratio of connections which are from v_i to vertices inside and outside the same set, respectively. The relaxed optimization of equation 3.4 is obtained by discarding the discreteness condition, but allowing y to take arbitrary real values. According to Rayleigh-Ritz theorem [207], the eigenvector corresponding to the second smallest general eigenvalue of $\mathbf{L}$ is the real valued solution to the relaxed version of equation 3.4. Finally, in order to partition the graph one can perform a simple thresholding on this eigenvector. The multi-class partitioning is also discussed in [292], where an iterative process of 2-way partition is implemented on the graph until a satisfactory result is achieved.

In fact, not limited to image segmentation, there has been several existing works in spectral graph clustering referring to the "graph cut" problem. The ra-

tio cut [158] and MinMaxCut [98] define different cut functions on the other types of data and lead to different graph Laplacians for clustering. All these methods overcome the drawback of Wu et al.'s minimal cut criterion and achieve "balanced" partitions. As a clustering method, spectral clustering often outperforms the traditional approaches in its efficiency and simplicity of implementation.

Let P be a directed path in G that starts and finishes at the same node v. Denote by $cost(P)$ the length of the boundary and by $weight(P)$ the segment-area cost. The graph cut cost function is the defined as:

$$Regioncut(A, B) = \frac{cost(P)}{weight(P)} \tag{3.5}$$

Obviously, this cut criterion favours large objects in the image and the object characteristic of smoothness is induced via the area and perimeter measures. This definition is very similar to 3.3 except that it is defined on a single region. Additionally, one can use different interior information such as the intensity, texture or the size of the region in coding the area term ($weight(P)$). The limitation of this method is that it can only segment enclosed objects due to the definition of cost function.

The mean cut [323] proposed by Wang et al. addresses the problem by defining an edge-weight function:

$$Meancut(A, B) = \frac{cut(A, B|w(u, v))}{cut(A, B|1)} \tag{3.6}$$

where $cut(A, B|w(u, v))$ is the cut cost between region A and region B given the edge weight $w(u, v)$, $cut(A, B|1)$ is defined similarly with all edge weights to be 1. This cut function minimizes the average edge weight in the cut boundary. It allows both open and closed boundaries and guarantees that partitions are connected. However, the mean cut criterion does not explicitly introduce the bias on the preference for large object regions or smooth boundaries. The authors argued that this lack of bias allows producing segmentations that are better aligned with image edges. The global minimization is performed in a polynomial time by graph theoretic algorithm, but limited to connected planar graphs. To solve the cost function equation 3.6 there are three reductions in their algorithm: from minimal mean cut to minimal mean simple cycle, from minimal mean simple cycle to negative simple cycle and from negative simple cycle to minima weight

perfect matching. Afterwards Wang and Siskind extended the mean cut to a more general form called ratio cut [323]. The ratio cut inherits the merit of mean cut, but corresponds to the average affinity per unit length of segmentation boundary instead of the average affinity per element of the cut boundary. Furthermore, graph vertices in ratio cut method correspond to regions which are created by iterated region-based segmentation. The cut function is formulated as:

$$Meancut(A, B) = \frac{cut_1(A, B)}{cut_2(A, B)} \tag{3.7}$$

where $cut_1(A, B)$ and $cut_2(A, B)$ are defined on the graphs of different iterations. Mean cut is the same as ratio cut when $cut_2(A, B)$ contains the unit weights. Minimization of ratio cut for arbitrary graph is NP-hard and thus the same reduction process is used as the mean cut.

Graph cut methods provide well-defined relationships between the segments while the problem of finding a cut in an arbitrary graph may be NP-hard. Efficient approximation of the solution needs to be studied. Since these methods form good basis for general image segmentation problems, they can be combined with other segmentation techniques for further extension.

3.2.3 Graph Cut on Markov Random Field Models

The study of psychology suggests that the use of contextual constraints is crucial for interpreting visual information. The Markov Random Field (MRF) theory provides a useful and consistent way of modelling contextual information. In this framework, the mutual influences among pixels can be formulated into conditional MRF distributions. The joint distribution of a MRF can be transformed into a simple form due to the equivalence between MRF's and Gibbs distributions. In conjunction with the Bayesian maximum a posterior (MAP) estimation, the MAP-MRF framework [136, 129, 137, 332, 135] formulates the labelling problem into a problem of minimizing an energy function: $f* = argmin_f E(f|d)$, where d is the observation of image elements, f is the unknown labelling and $E(f|d)$ is thus the posterior energy function. Compared with the graph cut methods, these methods tend to explicitly incorporate any desirable high-level contextual information in the energy function.

3.2.3.1 Bi-labelling Graph Cut Methods

Strategies for optimizing the energy functional can be various. From those defined on discrete set of variables, the combinatorial min-cut/max-flow graph cut algorithm [42] is a prominent one. Greig et al. [154] are the first to find out that powerful min-cut/max-flow algorithms can be used to minimize certain energy functions in image restoration. The energy functional they used is:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{(p,q) \in N} V_{p,q}(f_p, f_q) \tag{3.8}$$

where f_p is the label of an image pixel, $D_p(\cdot)$ is the regional term that measures the penalties for assigning f_p to p, $V_{p,q}(\cdot)$ is the boundary term for measuring the interaction potential and N is the neighbourhood set. This graph energy functional was used later to solve the multi-camera stereo problem [268] and further generalized to image segmentation for convex or non-convex problems.

The graph cut energy functional encodes both the constraints from user interaction and the regularization of the image smoothness under the MAP-MRF framework. In the graph cut model, edges E consist of two types of links to formulate these two constraints: $t-links$ and $n-links$. Visual terminal nodes are added in the graph to represent the user input information. For example, if one attempts to partition an image into two classes (i.e. the object and the background), the class information is then modelled as two visual terminal nodes based on the user input. With this setting, each node is connected to the terminal nodes by $t-links$ and each pair of neighboring nodes is connected by an $n-links$.

The work in [186] studies what energy functionals can be minimized via graph cut. In particular, it provides a simple necessary and sufficient condition on energy functionals of binary variables with double and triple cliques. The global optimal solution of minimal cut can be found by different combinatorial min-cut/max-flow algorithms [123, 142, 44, 164], where Boykov and Kolmogorov's augmenting-path based algorithm has the best performance for common vision problems [44]. For huge 2D or 3D grids, the parallelizing of graph cut algorithm has also been studied [298, 202, 92].

The most typical way to represent the object/background models is based on the intensity distributions (e.g. histogram). Blake et al. [34] suggested using

a Gaussian Mixture Model (GMM) to approximate the distributions. As the object/background models are updated interactively, the high-level contextual information is enhanced for a stable representation of the objects of interest. A similar way of iteratively updating the regional term was proposed in [252], where the information is obtained progressively from the local image. In each iteration only the local neighbouring regions to the labelled regions are involved in the optimization so that strong interference from the far unknown regions can be significantly reduced.

The boundary term of equation 3.8 reflects the smoothness of the segmentation and hence the penalty of neighbouring graph elements will be small if they are similar. To describe such a penalty, local intensity gradient or colour histograms are the most commonly used criteria. Boykov et al. [43] investigated geometric properties of segments. They showed that discrete topology of graph cut can approximate any continuous Riemannian metric space. Thus many of the well-known geometric methods based on level sets [239, 54, 238] can be also studied in the discrete space by combinatorial graph cut.

3.2.3.2 Multi-labelling Graph Cut Methods

The standard s/t graph cut algorithm can find the exact optimal solution for certain classes of energy functionals [186]; however, in many cases the number of labels for assigning to graph nodes is more than two and the minimization of energy functions becomes NP-hard. For approximate optimization Boykov et al. [46] developed the $\alpha - expansion - move$ and $\alpha\beta - swap - move$ algorithms to deal with multi-labelling problems for more general energy functionals. Although the algorithm can only find local minimum solutions, their effectiveness has been validated by extensive experiments. This work inspires more studies to incorporate various constraints in the energy functional. In [**?**, **?**, **?**] the authors used ordering constraints in object segmentation. By defining the spatial relationship between the objects, the impossible segmentation is ruled out. The improved $\alpha - expansion - move$ algorithms make the optimization of energy functional more effective under the constraints.

3.2.3.3 Graph Cut with Shape Prior

Incorporating the shape prior in graph cut has been proven very useful for image segmentation. This visual cue can be added in either the regional term or the boundary terms to force the segmented object to follow a certain pre-defined shape. The idea of using a signed distance map function to represent some shape was proposed by Kolmogorov and Boykov [186], where they pointed out that combining geometric concept of flux and length/area in the regional term can improve the segmentation quality of long thin objects. In [45] the gradient flow evolution of a surface was computed by the L_2 distance of the drifting from its current position. It guarantees that the shape is not very far from the previous position in the evolving process. Freedman et al. [127] used a similar idea as in level-sets [84, 83] to specify the template as a distance function whose zero level set corresponds to the template. The rigid and the scale transformations were also considered in this work, where the shape term is integrated into the boundary term of the energy functional. Instead of using the specific shape template, Das et al. [87] and [316] studied more generic shape priors for image segmentation. These shapes are defined on the relative positions of neighbouring pixel pairs, thus the neighbourhood system for incorporating the shape constraints is the same as for the boundary constraints. In 2-labelling case, minimizing the shape based energy functionals can be accomplished exactly with a graph cut if all the pairwise terms are submodular.

3.2.3.4 Interactive Graph Cut Methods

The interactive property of graph cut allows an efficient editing of segmentation results. The lazy snapping [198] and Grabcut [267] provide quick object marking schemes for better user experience. Users are allowed to loosely position seed points to indicate which parts of the image are objects and modify the segmentation results by editing the boundary with some soft constraint [198]. Alternatively, instead of putting seeds in both the object and background, a user can simply drag a rectangle around the candidate object to indicate the background region only [267]. The "incomplete labelling" leads to a considerable reduction of user interaction. In **Grabcut**, the graph cut algorithm is performed iteratively with an updating process on the object and background models. Lempitsky et al [195] method used a rectangle to impose the topological prior on the segmented

object. The prior is incorporated into the energy minimization framework which leads to an NP-hard integer programming. An approximated solution is achieved by first relaxing it to a convex continuous optimization problem and then using a new graph cut based algorithm as a rounding procedure for the original problem. A more advanced user interactive tool was developed by Liu et al. [203], called "Paint Selection". It provides instant feedback to the users when they drag the mouse. This progressive selection algorithm is implemented based on multicore graph-cut and adaptive band upsampling. Experiment shows that a series of local optimization guarantees the segmentation quality since in each step the algorithm will match users' directions as much as possible.

3.2.4 Shortest Path Based Methods

Finding the shortest path between two vertices is a classical problem in graph theory. In a weighted graph, the shortest path will connect the two vertices with minimized sum of edge weights. Formally, let s and t be two vertices of a connected weighted graph G. The goal is to find a path from s to t whose total edge weights is minimal. This is a single pair shortest path problem and there are several algorithms to solve it. The most well-known one is Dijkstra's algorithm [97, 96] based on dynamic programming. This algorithm is to grow a Dijkstra tree, starting at the vertex s, by adding at each iteration a frontier edge whose non-tree endpoint is as close to s as possible. After each iteration the vertices in Dijkstra are those to which the shortest paths from s have been found [156]. In the shortest path based image segmentation the problem of finding the best boundary segment is converted into finding the minimum cost path between the two vertices.

The livewire method [115, 114] allows the user to select an initial point on the boundary. The subsequent point is chosen such that the shortest path between the initial point and the current cursor position will best fit the object of interest. In this setting the boundary is represented as a sequence of oriented pixel edges. Each oriented edge carries a single cost value to measure the quality of boundary. The boundary wraps around the object at a real-time speed. Compared with tedious manual tracing, livewire provides a more accurate and more reproducible tool for segmentation task. The difficulty with livewire is that user has to accurately put the seeds near the desired boundary. When there is texture

or weak boundary, a lot of guidance from the user may be required to obtain an acceptable segmentation. Livewire requires a searching over the whole graph for the shortest paths, therefore a large amount of computational resource is needed when segmenting high resolution images. Live Iane [115] overcomes this limitation by confining the searching space in a much smaller range (5-100 pixels) and largely reduces the computational time in most cases. As a matter of fact, the use of shortest path in edge and contour detection has been investigated for many years. Early work in this area [210, 211] tried to improve the computing time by heuristic search methods. However, the computing time is still dependent on the amount of noise in the picture. Other works [221, 257] embed certain restrictions on the form of the contour, which are useful in specific applications. One recent work was proposed by Falcao et al. [113] who exploited some known properties of graphs to avoid the unnecessary shortest path computation and proposed a fast algorithm called live-wire-on-the fly. The acceleration of graph searching is based on the fact that the results of computation from the selected point can make use of the previous position of the cursor. Their algorithm has the advantage that there is no restriction on the shape or size of the boundary and that it is oriented so that its inner and outer parts are well defined. The latter would be very useful when there are stronger boundaries nearby. The same idea has been adopted by other segmentation methods such as graph cut based algorithms. A very similar technique called Intelligence Scissors [227, 228] integrates the boundary cooling and on-the-fly-training in the graph searching process and as a result it reduces the amount of user interaction and makes the boundary adhere to specific type of edges.

Bai et al. [22] used geodesic distance to assign the path weights and study the image segmentation under a different framework. Instead of computing the shortest path on the boundary, their algorithm is based on image regions. A pixel is assigned with a foreground label if there is a shorter path from that pixel to a foreground seed than to any background seed. The algorithm can be implemented very efficiently as the time complexity for geodesic is linear time. However, it is strongly dependent on the seed locations and is more likely to leak through weak boundaries.

Due to the increasing applications of 3D data in practice, researchers have been looking for the 3D extensions of the 2D shortest path techniques. 3D examples of live wire were proposed in [112, 160] for medical image segmentation.

Other 3D extensions of the shortest path algorithm can be found in [15, 16]. However, these extensions are not straightforward and fundamentally path-based techniques. There is no guarantee that the shortest paths will lie on the minimal surface. To solve this problem, Grady [149] adopted a mathematically elegant method to find the minimal surfaces and then used them to segment the 3D data.

In comparison with the MST based methods, which focus on the clustering properties of a segment, the shortest path can well describe certain nature of the object boundaries in the image. By virtue of its computational reliability, the image segmentation problem can be solved intuitively and effectively. Unlike contour evolution methods (e.g. active contour [80, 63]), live wire is based on a user-driven process where image features are used to define the graph model. In most circumstances live wire provides more freedom for user to control the segmentation process. It might be more suitable for extracting complex objects with relatively explicit boundaries than other graph based methods.

3.2.5 Other Methods

The random walker [147] is an interactive segmentation method that is formulated on a weighted graph to assign a label to each pixel on an image. Each edge on the graph is assigned a real valued weight defined as: $w_{ij} = \exp(-\beta(g_i - g_j)^2)$, where g_i is the image intensity at pixel i and β is a free parameter. This weight can be taken as the likelihood that a random walker will go across that edge. As a consequence, the label of a pixel is given by the seed point that the random walker first reaches. The theoretical basis of random walker is an analogue of the discrete potential theory on electrical circuits [102]. The solution of random walker probabilities has been found the same as minimizing a combinatorial Dirichlet problem [173]:

$$D[x] = \frac{1}{2} \sum_{e_{i,j \in E}} w_{ij}(x_i - x_j^2) \tag{3.9}$$

Minimizing $D[x]$ equals to solving the harmonic function that satisfies the boundary condition, which can be set by letting the seed point value be unit. The equation 3.9 has an identical form to graph cut function in equation 3.3.1.1.

Sinop et al. [294] unified the graph cuts [42] and random walker [147] into a general framework, which is based on the minimization of l_q norms. A new algorithm was therefore derived in the case of $q = \infty$.

Pavan et al. [244] proposed an image segmentation method based on dominant sets, which is a generalization of maximum clique in the context of weighted graph. The maximal clique is the strictest definition of a cluster [20], since it defines a cluster in the edge-weighted graph which has the internal homogeneity and the external inhomogeneity simultaneously. The dominant sets are converted into a continuous quadratic optimization problem and thus solved by the replicator dynamics from evolutionary game theory [324]. However, to compute the dominant set in a graph there is a requirement of comparing all possible pairs of pixels in an image. To reduce the computational load, an efficient solution to this problem was studied in [245]. The dominant sets clustering method has been proved with better classification performance in intensity, colour, texture image segmentation and it is competitive with other spectral graph clustering methods such as normalized cut method [292] in both clustering quality and computational cost.

3.3 Variational Methods

In recent years algorithms based on the graphical framework have gained in popularity as highly competitive for problems involving high dimensional data, with applications ranging from image processing to machine learning. The graphical setting has even been extended to deep learning techniques. In particular, [287] introduce the concept of graph convolutional recurrent networks, [50] compare them with graph convolutional neural networks. Other works on deep learning methods on graphs include [196, 222, 91]. The success of such methods is partly due to the many advantages offered by using a graph-based approach. First, graphical techniques not only provide useful information about the overall structure of the data, but also about the connections between pieces of data, via a similarity graph, composed of nodes and edges, where the nodes correspond to elements of the data and the edge weights encode the degree of similarity between pairs of vertices. Second, the graphical structure provides a general way to incorporate diverse types of data, such as hyperspetral data, text data, LIDAR data, images, video, etc. Methods for segmenting data can be classified into several categories: unsupervised, semisupervised and supervised algorithms. In *unsupervised* learning the algorithm is not provided with any training data

and it proceeds to obtain the structure and labels of the data without any prior information. In *semisupervised* learning, the method is equipped with small set of labelled data and the goal is to propagate the labels towards the unlabelled elements. *Supervised* approaches involves using a large training set, obtaining the parameters for the model and testing the model on a small testing set. One of the most common and basic methods for segmenting data is spectral clustering, an unsupervised algorithm which clusters the data according to the $K - means$ method applied to a small set of eigenfunctions of the graph Laplacian matrix [319]. The procedure can be used along with the Nyström extension method to obtain a $O(n)$ low-rank approximation of the graph Laplacian matrix, where n is the number of elements in the data.

An optimization technique for data classification used in the context of machine learning often involves minimizing a general form of energy (or cost) functional:

$$E(u) = R(u) + \mu||u - \hat{u}||_b^p$$

where u is the classification function, $R(u)$ is a regularization term and $||u - \hat{u}||_b^p$ is a fidelity term, incorporating most (supervised) or just a few (semisupervised) of the known values $\hat{u}$. In the case of unsupervised classification, the second term is replaced by a region homogeneity term, the first term is equipped with a penalty, or additional constraints are imposed. The type of regularization term $R(u)$ has nontrivial consequences for the final classification accuracy. When choosing the regularization norm, it is important to conserve the sharp discontinuities that may appear in the boundaries between classes. The total variation seminorm and Mumford-Shah energy are particularly successful in the latter task. Overall, [41] describe the structural properties of certain solutions of 3.3. Moreover, the authors of [310] analyse ℓ_1 vs ℓ_2 regularization for the resolution of ill-posed linear inverse and compressed sensing problems. In addition, [311] provide a definition and analysis of a generalized total variation seminorm. Recently, the Ginzburg-Landau (GL) energy has been explored as a regularization term for tasks such as semisupervised classification and image processing, since it serves as an approximation to total variation. Given a phase field variable u, the GL energy, introduced for the Euclidean space in the last century, involves competition between the convex functional $\int(\nabla u)^2 dx$ that induces smoothing,

with a double-well function $\int W(u)dx$, that separates its argument into phases. Another multiphase image segmentation method, based on a phase transition model of Modica and Mortola in material sciences, is described in [172]

In this section, several variational graph-based optimization methods are described. Subsection 3.3.1 outlines recent methods for image segmentation and data classification using the Ginzburg-Landau (GL) and Mumford-Shah functionals on graphs based on the Merriman-Bence-Osher (MBO) scheme. Subsection 3.3.2 provides an introduction to recent total variation (TV) algorithms for machine learning, while subsection 3.3.3 describes how the techniques described in **??** can be extended from image segmentation to data classification. In particular, it provides an alternative to MBO scheme for applying Mumford-Shah energy to data clustering

3.3.1 Merriman Bence Osher Scheme

This subsection describes a fast algorithm for a variational method in a graph setting. The method, which was developed by Bertozzi et al., is inspired by diffuse interface models and is based on spectral graph theory.

In their work [29] Bertozzi and Flenner introduce a graph-based model based on the Ginzburg-Landau functional. They propose a binary classification algorithm based on the minimization of the GL functional with a fidelity term,

$$E(u) = \frac{\epsilon}{2}\int |\nabla u|^2 dx + \frac{1}{\epsilon}\int W(u)dx + F\left(u, u_0\right)$$

where u_0 is the initial state of the system. To define the functional on a graph, they replaced the spatial gradient by a more general graph gradient operator: the $\frac{\epsilon}{2}\int |\nabla u|^2 dx$ term is replaced with the more general graph operator term $\epsilon u \cdot L_s u$, so that

$$E(u) = \epsilon u \cdot L_s u + \frac{1}{\epsilon}\int W(u)dx + \int F\left(u, u_0\right)$$

The functional is minimized using the method of gradient descent, resulting in the following expression:

$$\frac{\partial u}{\partial t} = -\epsilon L_s u - \frac{1}{\epsilon}W'(u) - \frac{\partial F}{\partial u}$$

that this is the Allen-Cahn equation with a fidelity term, where Δu is replaced by a graph operator term $-L_S$. Taking F to be $\frac{1}{2}C\lambda(x)(u-u_0)^2$ for some constant C, one obtains

$$\frac{\partial u}{\partial t} = -\epsilon L_s u - \frac{1}{\epsilon}W'(u) - C\lambda(x)(u-u_0) \tag{3.10}$$

The main purpose of this section is to describe a fast and simple method for solving 3.10 in the small ϵ limit. This is based on a graph-based Merriman-Bence-Osher (MBO) scheme, which uses simple threshhold dynamics to approximate motion by mean curvature. Since the Allen-Cahn equation is closely related to motion by mean curvature, this scheme has been very successful in solving different its variants. For example, the authors of [108] propose an adaptation of the MBO scheme to minimize the piecewise constant Mumford-Shah functional. Inspired by this work, Bertozzi et al. decide to adapt it for solving 3.10. However, the implementation of the proposed scheme poses many computational challenges. The quadratic size of graph Laplacian could make the iterative process very computationally expensive. To reduce the dimension of the graph Laplacian and make the computation more efficient, Bertozzi et al. propose the Nyström extension method for approximating eigenvalues and the corresponding eigenvectors.

3.3.1.1 Background

Numerous image segmentation energy functionals use a binary segmentation function that takes a certain value inside the segmented region and a different one outside of the segmented region. In their pioneering work [229], Mumford and Shah propose an energy functional that uses the perimeter of the segmentation function as regularizer. Many papers, such as [72], successfully use the total variation (TV) as seminorm

$$\|u\|_{TV} = \int_\Omega |\nabla u| dx$$

to approximate the perimeter of the front between the two values of the segmentation function. As an alternative to this approach, some researchers , such as Esedoglu and Tsai in their work [108], use the Ginzburg-Landau functional

$$GL(u) = \frac{\epsilon}{2}\int |\nabla u|^2 dx + \frac{1}{\epsilon}\int W(u) dx$$

to approximate the perimeter of the front. In this case, $W(u) = (u^2 - 1)^2$ and is a double well potential. A proof in [183] shows that the perimeter is the limit in the sense of Γ-convergence of the Ginzburg-Landau functional. Therefore, one can write

$$GL(u) \rightarrow_\Gamma C|u|_{TV}$$

This convergence allows the two functionals to be interchanged in some cases. One might prefer to use the Ginzburg-Landau functional instead of the TV semi norm since its highest order term is purely quadratic, which allows for efficient minimization procedures. In contrast, minimization of the TV semi norm leads to a nonlinear curvature term, making it less trivial to solve numerically. However, recent advances, such as the split Bregman method described in [143], have made progress in such problems. Due to its connection to the TV semi norm, the Ginzburg-Landau functional has also often been used in image processing and in various image processing applications, such as inpainting [99, 28] and segmentation [109, 108]. In practice, one would minimize

$$E(u) = GL(u) + F(u, u_0)$$

where F is the fidelity term and u_0 is the initial state of the system. When one minimizes the Ginzburg-Landau functional, the function u approaches either one of the two minimizers, 1 and -1. of the double well potential. However, the presence of the gradient term will force u to be somewhat smooth, i.e., without any sharp transitions between 1 and -1. Therefore, the function that minimizes the functional will have regions where it is close to -1, regions close to 1, and a thin region of scale $O(\epsilon)$ which is somewhere in between. Since the minimizer appears to have two phases with an interface between them, models involving the Ginzburg-Landau functional are typically referred to as "diffuse interface models".

In order to understand the minimization algorithm designed by Bertozzi et al. it is useful to recall some useful notions in graph theory, described in [78]. Consider an undirected graph $G = (V, E)$, where V and E are the sets of vertices and edges, respectively. The vertices are, for example, points in $\mathbb{R}^n$ or pixels in a image. Let w be the weight function, where $w(i, j)$ represents the weight (often measured between 0 and 1) between vertices i and j and $w(i, i)$ is set to zero. The

weight represents a measure of similarity between the vertices; thus, two vertices having a weight close to 1 are very similar to each other, and two vertices having a weight close to 0 are dissimilar. Now, let the degree of a vertex $i \in V$ be defined as

$$d(i) = \sum_{j \in V} w(i,j)$$

Using the above, one defines the graph Laplacian to be the matrix L such that

$$L(i,j) = \begin{cases} d(i) & \text{if } i = j \\ -w(i,j) & \text{otherwise} \end{cases}$$

If one defines the degree matrix D to be the $N \times N$ diagonal matrix with diagonal elements $d(i)$, then the graph Laplacian can be written in matrix form as $L = D - W$, where W is the matrix $w(i,j)$. The matrix W is sometimes referred to as the "affinity matrix". The graph Laplacian satisfies the equations

$$Lu(i) = \sum_{j} w(i,j)(u(i) - u(j))$$

$$u \cdot Lu = \frac{1}{2} \sum_{i,j} w(i,j)(u(i) - u(j))^2$$

for all $u \in \mathbb{R}^n$ and has nonnegative, real-valued eigenvalues, including 0. When working with the graph Laplacian, one must consider the behaviour that arises as the sample size grows larger. Increasing sample size leads to decreasing grid size; thus, the operator must be scaled to converge to the differential Laplacian as $N \to \infty$, where N is the number of vertices. Although several versions that have been shown to have the correct scaling in the limit exist, the one used here is the symmetric Laplacian

$$L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

that satisfies

$$u \cdot L_s u = \frac{1}{2} \sum_{i,j} \frac{w(i,j)(u(i) - u(j))^2}{\sqrt{d(i)d(j)}} \quad \forall u \in \mathbb{R}^n$$

Bertozzi et al. use this version since the symmetric property of the matrix allows for more efficient algorithms for calculating eigenvectors. Another version that is commonly used is the random walk Laplacian,

$$L_w = D^{-1}L = I - D^{-1}W$$

which is related to Markov processes. More detail about Laplacians is given in [78] and [319].

As mentioned in previously the weight function $w(i, j)$ is a function that measures the degree of similarity between vertices i and j. Therefore, it is necessary to choose the function in such a way that two vertices that are heavily weighted by w, i.e., $w(i, j)$ large, are also closely related in the data. Although several options for w are discussed in [319], the choice depends on the problem, so no general theory can be formulated. One popular choice for the similarity is the Gaussian function

$$w(i, j) = e^{-\frac{d(i,j)^2}{\sigma^2}}$$

where $d(i, j)$ is some distance measure between the two vertices i and j, and σ is a parameter to be chosen. Von Luxurg in [319] explains that σ can be chosen to be on the order of $log(N) + 1$, where N is the number of vertices. This similarity function is an appropriate choice when vertices are, for example, points in $\mathbb{R}^n$, since two points that are close together are more likely to belong to the same cluster than are two points that are far apart. The choice of $d(i, j)$ varies with the data set. If one wants to cluster points in $\mathbb{R}^n$, a reasonable choice for $d(i, j)$ is the Euclidean distance between points i and j. In the case of image processing, where the vertices are the pixels in the image, the concept of feature vectors is used to construct $d(i, j)$, as in[29]. Each vertex i is assigned an n-dimensional feature vector, and $d(i, j)$ is then the weighted 2-norm (where each coordinate of the vector is assigned a weight) of the difference of the feature vectors of pixels i and j. The goal of graph clustering is to partition the graph so that the weights between vertices of different groups are small and the weights between vertices within the same group are large. A mincut approach to the above problem is to partition a set of vertices V into sets A and $\overline{A}$ in such a way that

$$\text{cut}(A, \bar{A}) = \sum_{x \in A, y \in \bar{A}} w(x, y)$$

is minimized. This mincut problem can be solved using an efficient algorithm like the ones in [297, 175, 176]. However, this problem leads to poor classification in many cases since the resulting "bad" partition often isolates one vertex from the rest of the set [219]. One way to overcome this problem is to use correct normalization, i.e., to force the sets A and $\overline{A}$ to be "large". Let

$$\text{vol}(A) = \sum_{x \in A} d(x)$$

then the modified problem is to find a subset A of V such that

$$\text{Ncut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{\text{vol}(A)} + \frac{\text{cut}(A, \bar{A})}{\text{vol}(\bar{A})}$$

is minimized. This is a NP-hard problem [322]. One way to simplify it would be to allow the solution to take arbitrary values in $\mathbb{R}$. This leads to the following relaxed NCut

$$\min_{A \subset Y} \langle u, L_s u \rangle, \quad u \perp D^{\frac{1}{2}} 1, \quad \|u\|^2 = \text{vol}(Y)$$

The fact that the above problem obtains a real-valued solution instead of a discrete-valued solution, like problem 3.3.1.1, is emphasized. The relaxed problem 3.3.1.1 has been applied to many segmentation problems; for example, appealing results are shown in [292]. To solve the above problem, one can apply the Rayleigh-Ritz theorem, and the solution is given by the second eigenvector ot the symmetric graph L_s [319].

In the following, fully connected graphs and non-local operators are used because they allow to capture patterns and texture in the image by using non-local information.

Spectral Clustering [319] is a popular approach for clustering data set into several classes. The method requires the data set to be embedded in a graph framework and the eigenvectors of the graph Laplacian (or the random walk Laplacian) to be computed.

The K-means [208] algorithm for finding K clusters proceeds iteratively by first choosing K centroids and then assigning each point to the cluster of the

Algorithm 3: Spectral Clustering

Data: Graph Laplacian $L(or L_w)$, number K of clusters to construct.

Result: Clusters $A_1 \dots A_K$ with $A_i = \{j | y_j \in C_i\}$

1 Compute first K eigenvectors $v_1 \dots v_K$ of L (or L_w);

2 Let $V \in \mathbb{R}^{N \times K}$ be the matrix containing the vectors $v_1, \dots, v_K$ as columns;

3 For $i = 1, \dots, N$ let $y_i \in \mathbb{R}^{\mathbb{K}}$ be the vector corresponding to the $i^{th}\ row of\ V$;

4 Cluster the points $(y_i)_{i=1,\dots,N}$ with the $K-$means algorithm into clusters $C_1, \dots, C_K$;

nearest centroid. The centroid of each cluster is the recalculated and the iterations continue until there is little change from one iteration to the next. A generalized version of spectral clustering using the $p - Laplacian$ is proposed in [53].

In general, image processing methods that are local fail to produce satisfactory results on images with repetitive structures and textures because they operate only on small neighbourhoods, without using any information about the whole domain. The advantage of non-local operators is that they contain data about the whole vertex set and are thus more successful with those types of images. Zhou and Schölkopf in their papers [343] formulated a theory of non-local operators that is related to the discrete graph Laplacian. Gilboa and Osher proposed using non-local operators to define functionals involving the TV seminorm for various image processing applications in their work [139]. A review of non-local calculus is presented below. Let $\Omega \in \mathbb{R}^n$, $u(x)$ be a function $u : \Omega \to \mathbb{R}$ and the non-local derivative be defined as

$$\frac{\partial u}{\partial y}(x) = \frac{u(y) - u(x)}{d(x, y)}, \quad x, y \in \Omega$$

where d is some positive distance defined on the space and $0 < d(x, y) < \infty$ for all x, y. If the (symmetric) weight function is defined as

$$w(x, y) = \frac{1}{d(x, y)^2}$$

the non-local derivative can be written as

$$\frac{\partial u}{\partial y}(x) = (u(y) - u(x))\sqrt{w(x, y)}$$

Vectors are denoted by $\vec{v} = v(x, y) \in \Omega \times \Omega$. Let $\vec{v}_1$ and $\vec{v}_2$ be two such vectors.

The dot product and the inner product are defined as

$$(\vec{v}_1 \cdot \vec{v}_2)(x) = \int_\Omega v_1(x,y)v_2(x,y)dy$$

$$\langle \vec{v}_1, \vec{v}_2 \rangle = \langle \vec{v}_1 \cdot \vec{v}_2, 1 \rangle = \int_{\Omega\times\Omega} v_1(x,y)v_2(x,y)dxdy$$

The magnitude of a vector can be defined as

$$|v|(x) = \sqrt{\vec{v} \cdot \vec{v}} = \sqrt{\int_\Omega v(x,y)^2 dy}$$

while the non-local gradient $\nabla_w u(x) : \Omega \to \Omega \times \Omega$ is the vector of all partial derivatives:

$$(\nabla_w u)(x,y) = (u(y) - u(x))\sqrt{w(x,y)}, \quad x,y \in \Omega$$

With the above definitions, the non local divergence $\operatorname{div}_w \vec{v}(x) : \Omega \times \Omega \to \Omega$ is defined as the adjoint of the non-local gradient:

$$(\operatorname{div}_w \vec{v})(x) = \int_\Omega (v(x,y) - v(y,x))\sqrt{w(x,y)}dy$$

The Laplacian is now defined as

$$\Delta_w u(x) = \frac{1}{2}\operatorname{div}_w(\nabla_w u(x)) = \int_\Omega (u(y) - u(x))w(x,y)dy$$

Since the graph Laplacian was defined as

$$Lu(x) = \sum_y w(x,y)(u(x) - u(y))$$

one can interpret $-Lu(x)$ as a discrete approximation of $\Delta_w u$. Note that a constant of $\frac{1}{2}$ was needed here to relate the two Laplacians. According to the non-local calculus described above,

$$\int_\Omega |\nabla u|^2 dx = \int_{\Omega\times\Omega} (u(y) - u(x))^2 w(x,y)dxdy$$

since

$$u \cdot Lu = \frac{1}{2} \sum_{x,y} w(x,y)(u(x) - u(y))^2$$

one can consider $2u \cdot Lu$ to be the discrete version of $\int |\nabla u|^2 dx$. In their paper [29], Bertozzi and Flenner replace the $\frac{\epsilon}{2} \int |\nabla u|^2 dx$ term of 3.3.1.1 by $\epsilon u \cdot Lu(x)$. However, normalization of the Laplacian is necessary (refer to [29]), so instead they use

$$\epsilon u \cdot L_s u = \frac{\epsilon}{2} \sum_{x,y} \frac{w(x,y)(u(x) - u(y))^2}{\sqrt{d(x)d(y)}}$$

When the variational solution u takes the values -1 or 1.

$$u \cdot L_s u = C + 4 \sum_{x \in A, y \in \bar{A}} \frac{w(x,y)}{\sqrt{d(x)d(y)}} - 2 \left(\sum_{x \in A, y \in A} \frac{w(x,y)}{\sqrt{d(x)d(y)}} + \sum_{x \in \bar{A}, y \in \bar{A}} \frac{w(x,y)}{\sqrt{d(x)d(y)}} \right)$$

In this case, C is a constant that varies with the graph, but not with the partition. The representation shows that the above is minimized when the normalized weights between vertices of different groups are small, but the normalized weights between vertices within a group are large. This is precisely the goal of graph clustering. Therefore, by replacing the $\frac{\epsilon}{2} \int |\nabla u|^2 dx$ term of 3.3.1.1 with $\epsilon u \cdot Lu(x)$, thus creating a graph-based version of 3.3.1.1, and then minimizing the resulting equation, one achieves the desired segmentation. The Γ-convergence of the graph-based Ginzburg-Landau functional is investigated in [313]. The authors prove that as $\epsilon \to 0$, the limit is related to the TV seminorm and cut from 3.3.1.1. Another important operator that arises from the need to define variational methods on graphs is the mean curvature on graphs. This non-local operator was introduced by Osher and Shen in [242], who defined it via graph-based p-Laplacian operators. p-Laplacian operators are a family of quasi-linear elliptic partial differential operators defined for $1 \leq p < \infty$:

$$L^p(f) = \nabla \cdot \left(|\nabla f|^{p-2} \nabla f \right)$$

In the special case $p = 2$, the p-Laplacian is just a regular Laplacian. For $p = 1$, the p-Laplacian represents curvature. The discrete graph version of p-Laplace operators is defined in [106] as

$$L^p(u(x)) = \frac{1}{p} \sum_{(x,y)\in E} w(x,y) \left(\|\nabla u(x)\|^{p-2} + \|\nabla u(y)\|^{p-2} \right) (u(x) - u(y))$$

Note that the graph 2-Laplacian is just the graph Laplacian, which is consistent with continuous case. Let us now define the mean curvature on graphs, which is the discrete analogue of the mean curvature of the level of a function defined on a continuous domain of $\mathbb{R}^N$:

$$\kappa_w = \frac{1}{2} \sum_{(x,y)\in E} w(x,y) \left(\frac{1}{\|\nabla u(x)\|} + \frac{1}{\|\nabla u(y)\|} \right) (u(x) - u(y))$$

Note that in the case of an unweighted mesh graph, k_w becomes a numerical discretization of the mean curvature. This curvature, k_w is also used in [93] as a regularizer in a graph adaptation of the Chan-Vese method. In their work [314], van Gennip et al. propose a different definition of mean curvature on graphs and prove convergence of the MBO scheme on graphs.

3.3.1.2 Algorithms

The idea of approximating mean curvature flow using threshold dynamics was introduced in [214] by Merriman, Bence and Osher (MBO). They propose a new computational algorithm for tracking the evolution in time of a set in $\mathbb{R}^n$ whose boundary moves with normal velocity equaling ($(n-1)$ times) its mean curvature. The procedure is this. Given a compact set $C_0 \subset \mathbb{R}^n$ the heat equation is solved with initial data the indicator function of C_0:

$$\begin{cases} u_t - \Delta u = 0 & \text{in } \mathbb{R}^n \times (0, \infty) \\ u = \chi_{C_0} & \text{on } \mathbb{R}^n \times \{t = 0\} \end{cases}$$

Fix now a time step $t > 0$ and define the new set

$$C_t \equiv \left\{ x \in \mathbf{R}^n \mid u(x,t) \geq \frac{1}{2} \right\}$$

One can write

$$C_t = \mathcal{H}(t) C_0$$

where $t \geq 0$ and regard $\mathcal{H}(t)$ for each time $t > 0$ as a mapping on the collection $\mathcal{K}$ of compact subsets of $\mathbb{R}^n$. $\mathcal{H}(t)_{t\geq 0}$ represents *heat diffusion flow*. As explained heuristically in [215] the evolution of C_0 into $C_t = \mathcal{H}(t)C$ approximates for small times t the mean curvature motion of the boundary Γ_0 of C_0, at least if Γ_0 is smooth. Hence continually reinitiating the procedure over short time steps should yield an approximation to mean curvature flow, valid even for large times. From the previous discussion follows the MBO numerical scheme for approximation of the motion u by mean curvature at discrete times:

- Step 1 (diffusion). Let $v(x) = S(\delta t)u_n(x)$, where $S(\delta t)$ is the propagator (by time δt) of

$$\frac{\partial v}{\partial t} = \Delta v$$

- Step 2 (thresholding). Set

$$u^{n+1}(x) = \begin{cases} 1 & \text{if } v(x) \geq \frac{1}{2} \\ 0 & \text{if } v(x) < \frac{1}{2} \end{cases}$$

The minimization of Ginzburg-Landau functional by gradient descent yields the Allen-Cahn equation:

$$\frac{\partial u}{\partial t} = 2\epsilon\Delta u - \frac{1}{\epsilon}W'(u)$$

here W is the double well potential $W(u) = (u^2 - 1)^2$. It is proven in [269] that as $\epsilon \to 0^+$, the rescaled solutions $u_\epsilon(x, t/\epsilon)$ of the above equation move according to mean curvature of the interface between the -1 and 1 phases of the solutions. In addition, [23] and [111] present rigorous proofs that the MBO algorithm approximates motion by mean curvature. This implies that for the small values of ϵ, the MBO thresholding scheme can be used to numerically solve the Allen-Cahn equation. Multiple extensions, adaptations and applications of the MBO scheme are present in literature. [108] proposes a modification of the MBO scheme for solving the inhomogeneous Allen-Cahn equation. To create a fast image segmentation algorithm, Esedoglu and Tsai propose a thresholding scheme form minimizing a diffuse interface version of the piecewise constant Mumford-Shah functional, [108] propose a MBO scheme for minimizing the diffuse interface

version of the piecewise constant Mumford-Shah functional

$$MS_\epsilon(u, c_1, c_2) = \int_D \epsilon |\nabla u|^2 + \frac{1}{\epsilon} W(u) + \lambda \left\{ u^2 (c_1 - f)^2 + (1-u)^2 (c_2 - f)^2 \right\} dx \tag{3.11}$$

where f is the image. The first variation of the model 3.11 yields the following gradient descent equation:

$$u_t = 2\epsilon \Delta u - \frac{1}{\epsilon} W'(u) + 2\lambda \left\{ u (c_1 - f)^2 + (1-u)(c_2 - f)^2 \right\}$$

and the adaptation of the MBO scheme is used to solve it. Esedoglu and Tsai propose the following scheme (similar to the MBO scheme, where the propagation step based on the heat equation is combined with thresholding):

- Step 1 (diffusion) Let $v(x) = S(\delta t)$ is a propagator by time δt of the equation

$$w_t = \Delta w - 2\tilde{\lambda} \left(w (c_1 - f)^2 + (1-w)(c_2 - f)^2 \right)$$

  with appropriate boundary conditions.

- Step 2 (thresholding) Set

$$u_{n+1}(x) = \begin{cases} 0 & \text{if } v(x) \in \left(-\infty, \frac{1}{2}\right] \\ 1 & \text{if } v(x) \in \left(\frac{1}{2}, \infty\right) \end{cases}$$

Some other extensions of the MBO scheme appeared in [314, 110, 215]. An efficient algorithm for motion by mean curvature using adaptive rids was proposed in [277].

This method involves the computation of eigenvalues and associated eigenvectors of the symmetric graph Laplacian. In practice, one needs to compute only a fraction of the eigenvalues and eigenvectors (since eigenvectors with very small eigenvalues are not very significant computationally), and different methods of doing so are depending on the size of the domain. When the graph is sparse and is of moderate size, around 5000×5000 or less, Rayleigh-Chebyshev procedure can be applied [14]. When the graph is very large, such as in the case of image classification, the Nyström extension method is used. Nyström extension [124] is a matrix completion method often used in image processing applications, such

as kernel principle component analysis [27] and spectral clustering [124]. This procedure performs much faster the many alternate techniques because it uses approximations based on calculations on small submatrices of the original matrix. When the size of the matrix becomes very large, this method is especially valuable. A detailed description can be found in [213].

Bertozzi et al. designed several data classification algorithms based on the MBO scheme for minimizing 3.3.1. For small ϵ the MBO thresholding scheme can be used to evolve the Allen-Cahn equation to a steady state. The scheme consists of two steps: a heat equation propagation step and a thresholding step. Therefore, the algorithm consists of alternating between the following two steps to obtain approximate solutions $u_n(x)$ at discrete times:

- Step 1 (diffusion) Propagate using

$$\frac{\partial y}{\partial t} = -L_s y - C_1 \lambda(x)\,(y - u_0)$$

  starting with u_n.

- Step 2 (thresholding). Set

$$u^{n+1}(x) = \begin{cases} 1 & \text{if } y(x) \geq 0 \\ -1 & \text{if } y(x) < 0 \end{cases}$$

Bertozzi et al. discretize 3.3.1.2 above in the following manner:

$$\frac{u^{n+1} - u^n}{dt} = -L_s u^{n+1} - C_1 \lambda(x)\,(u^n - u_0)$$

The symmetric Laplacian is calculated implicitly. An implicit term is needed, since an explicit scheme requires all the scales of the eigenvalues to be resolved numerically. The above scheme is used because it is the simplest scheme possible keeping the Laplacian term implicit. The scheme is solved using the spectral decomposition of the symmetric graph Laplacian. Let $u^n = \sum_k a_k^n \phi_k(x)$ and $C_1\lambda\,(u^n - u_0) = \sum_k d_k^n \phi_k(x)$, where $\phi(x)$ are the eigenfuntions of the symmetric Laplacian. Using the obtained representation and 3.3.1.2 one obtains

$$\frac{u^{n+1} - u^n}{dt} = -L_s u^{n+1} - C_1 \lambda(x)\,(u^n - u_0)$$

where λ_k are the eigenvalues of the symmetric graph Laplacian. This spectral decomposition method is chosen because it is very efficient. Without it, the discrete Laplacian term by itself requires $O(N^2)$ calculations (without assuming any sparsity). However, when using spectral decomposition, one obtains the advantage of only having to calculate the first few largest eigenvalues and associated eigenvectors (as the smallest eigenvalues and associated eigenvectors become insignificant in calculations). Therefore, the discrete Laplacian term now requires only $O(NL)$ calculations, where L is the number of eigenvalues/eigenvectors calculated. Of course, this method is useful only if there is an efficient way to calculate the eigenvalues and eigenvectors of the symmetric Laplacian. Therefore, the algorithm consists of the following:

Step 1 Create a graph from the data, choose a similarity function, and then calculate the symmetric graph Laplacian.

Step 2 Calculate the eigenvectors and eigenvalues of the symmetric graph Laplacian. It is necessary only to calculate a portion of the eigenvectors.

Step 3 Initialize u.

Step 4 Apply the two-step scheme (to minimize the Ginzburg-Landau functional) described above for a certain number of iterations until stopping criterion is satisfied. Use the following method:

1. Let $u_k^0 = \sum_x u_0 \phi_k(x)$ and $d_k^0 = 0$ for all x
2. Until a stopping criterion is satisfied, do the following:
   a. Repeat for some number s of steps:
      1. $a_k^n \leftarrow \frac{a_k^n - \delta t d_k^n}{1+\delta t \lambda_k}$
      2. $y(x) = \sum_k a_k^n \phi_k(x)$
      3. $d_k^n = \sum_x C_1(y - u_0)(x)\phi_k(x)$
   b. (thresholding part)

$$u^{n+1}(x) = \begin{cases} 1 & \text{if } y(x) \geq 0 \\ -1 & \text{if } y(x) < 0 \end{cases}$$

   c. Let $a_k^{n+1} = \sum_x u_{n+1}(x)\phi_k(x)$ and $d_k^{n+1} = \sum_x C_1(y - u_0)(x)\phi_k(x)$

The parameter δt is chosen using trial and error. The used stopping criteria is $\frac{\|u_{new}-u_{\text{old}}\|_2^2}{\|u_{\text{new}}\|_2^2} < \alpha = 0.0000001$.

Semi-supervised Algorithm In semi-supervised learning, the fidelity, or a small amount of "ground truth", is known and the rest of the data set needs to be classified according to the categories of the known data. In this section the semi-supervised classification problem is approached by using energy minimization techniques. Similar approaches have been used in [29], where the problem is formulated as a minimization of the Ginzburg-Landau (GL) functional (in graph form) with a fidelity term. In [213], the authors propose an MBO scheme to solve the binary classification; a multi-class extension of that algorithm is described in [131, 29].

The problem is to classify a data set with N elements into $\hat{n}$ classes, where $\hat{n}$ is to be provided to the algorithm in advance. An assignment matrix u is an $N \times \hat{n}$ matrix, where each row is an element of the Gibbs simplex $\sum^{\hat{n}}$, defined as

$$\Sigma^{\hat{n}} := \left\{ (x_1, \ldots, x_{\hat{n}}) \in [0,1]^{\hat{n}} \mid \sum_{k=1}^{\hat{n}} x_k = 1 \right\}$$

Therefore, each row of u is a probability distribution: the k^{th} component of the i^{th} row of u is the probability the i^{th} node belongs to class k. In the text that follows, the i^{th} row of u is denoted by u_i. Let e_k be the k^{th} vertex of the simplex, where all the entries are zero, except the k^{th} which is equal to one. The optimization problem consists of minimizing the following energy

$$E(u) = \epsilon \langle u, L_s u \rangle + \frac{1}{\epsilon} \sum_i W(u_i) dx + \sum_i \frac{\mu}{2} \lambda(x_i) ||u_i - \hat{u}_i||_{L_2}^2 \tag{3.12}$$

The first two term of 3.12 comprise the graph form of the Ginzburg-Landau functional, where L_S is the symmetric Laplacian, ϵ is a small positive constant and $W(u_i)$ is the multi-well potential in $\hat{n}$, where $\hat{n}$ is the number of classes.

$$W(u_i) = \prod_{k=1}^{\hat{n}} \frac{1}{4} ||u_i - e_k||_{L_1}^2$$

The last term of 3.12 is the regular L_2 fit to known data with some constant

μ, while $\lambda(x)$ takes the value of 1 on fidelity nodes and 0 otherwise. The variable $\hat{u}$ is the initial value for u with randomly chosen labels for non-fidelity data points and the "ground truth for the fidelity points. Lastly, in 3.12 for matrices A and B, $\langle A, B\rangle = \text{trace}\left(A^T B\right)$, where A^T indicates A transpose. Minimizing $E(u)$ by the gradient descent method, one obtains

$$\frac{\partial u}{\partial t} = -\epsilon L_s u - \frac{1}{\epsilon} W'(u) - \mu\lambda(x)(u - \hat{u}) \tag{3.13}$$

This is the Allen-Cahn equation [4, 118, **?**] with fidelity term with the differential operator ∇u replaced by a more general operator $-L_S$ [206]; when $\epsilon \to 0$, the solution to the Allen-Cahn equation approximates motion by mean curvature [214]. It is important to note that in the last term of 3.13 the product is meant to be calculated on each node.

In [130], the authors propose an MBO scheme to solve 3.13. Here a semi-supervised algorithm is presented and is based on [130]. The initialization $\hat{u}$ is composed by known labels for the fidelity points and random class labels for non-fidelity points. To obtain the next iterate of u, one proceeds with the following two steps:

Step 1 Heat equation with forcing term:

$$\frac{u^{n+\frac{1}{2}} - u^n}{dt} = -L_s u^n - \mu\lambda(x)\left(u^n - \hat{u}\right)$$

Step 2 Threshhold

$$u_i^{n+1} = e_r, r = arg\ max\ u_i^{n+\frac{1}{2}}$$

for all $i \in 1, 2, \ldots, N$, where e_r is the r^{th} standard basis in $\mathbb{R}^{\hat{n}}$.

For a stopping criterion, one computes the norm of the difference between the label matrix u of two consecutive iterations and stop the iteration when the norm is below a threshold value. If one denotes the final u by u_f. To obtain the final classification of node i, the largest value in the i^{th} row of u_f is chosen and the corresponding index is assigned as the class label of node i. Step 1 can be computed very efficiently and simply by using the eigendecomposition of L_S, which is

$$L_S = X\Lambda X^T$$

where X is the eigenvector matrix and Λ is a diagonal matrix containing the eigenvalues. X can be approximated by a truncated matrix retaining only a small number of the leading eigenvectors. If one writes

$$u^n = Xa^n,\ \mu\lambda(x)(u^n - \hat{u}) = Xd^n$$

and equates coefficients, Step 1 can be formulated in the MBO scheme as solving for the coefficients

$$a_k^{n+1} = (1 - dt\lambda_k) \cdot a_k^n - dt \cdot d_k^n \tag{3.14}$$

where λ_k is the k^{th} eigenvalue of L_S in ascending order. Due to the fact that, in practice, only the leading eigenvalues and eigenvectors (in ascending order) need to be calculated to obtain a good accuracy, 3.14 is an efficient way to compute Step 1 of the algorithm, even in the case when th number of classes is very large. Empirically, the algorithm converges after a small number of iterations. The iterations stop when a purity score between the partitions from two consecutive iterations is grater than 99.99%. The purity score measures how "similar" two partitions are. Intuitively, it can be viewed as the fraction of nodes of one partition that have been assigned to the correct class with respect to the other partition.

Unsupervised algorithm In order to handle the case when there is no knowledge of the class of any part of the data set an unsupervised algorithm is presented. This method is based on the Mumford-Shah [229] (see 2.1.2.4). One simplified version of the Mumford-Shah model tailored for images is the piecewise constant model [73, 108]

$$E^{MS}(\Phi, \{c_r\}_{r=1}^{\hat{n}}) = |\Phi| + \mu \sum_{r=1}^{\hat{n}} \int_{\Omega_r} (f - c_r)^2, \tag{3.15}$$

where the contour Φ segments an image region Ω into $\hat{n}$ disjoint sub-regions Ω_r, $|\Phi|$ is the length of contour, f is the observed image data, μ is a constant and $\{c_r\}_{r=1}^{\hat{n}}$ is a set of constant values which represent the local centroids. The graph version of the multi-class piecewise constant Mumford-Shah energy was

introduced in [167] for hyperspectral images.

$$MS(u, \{c_r\}_{r=1}^{\hat{n}}) = \frac{1}{2}|u|_{TV} + \mu \sum_{r=1}^{\hat{n}} \langle \|f - c_r\|^2, u_{*,r} \rangle, \tag{3.16}$$

where u is the class assignment matrix in which each row is an element of the Gibbs simplex. The length of the contour is estimated by total variation (TV) of the assignment matrix u. In 3.16 the term $||f - c_r||^2$ denotes an $N \times 1$ vector $(||f(x_1) - c_r||^2, \ldots, ||f(x_N) - c_r||^2)^T$ and x_i $(i = 1, \ldots, N)$ are the N pixels of the data set. In addition, the term $u_{\star,r}$ indicates the r^{th} column of u: the vector $u_{\star,r}$ is a $N \times 1$ vector which contains the probabilities of every node belonging to class r. Lastly, in 3.16, $\langle , \rangle$ indicates the usual inner product. The problem is to classify a data with N elements into $\hat{n}$ classes, where $\hat{n}$ is to be provided to the algorithm in advance. u is the assignment matrix, defined in the previous section and one needs to minimize 3.16. This problem is essentially equivalent ot the K-means method when μ approaches $+\infty$. Minimizing the variation in c yields the following formula for the optimal constants c_r

$$c_r = \frac{\langle f, u_r \rangle}{\sum_{i=1}^{N} u_{\star,r}(x_i)}$$

where $u_{\star,r}(x_i)$ indicates the i^{th} entry of $u_{\star,r}$.

This algorithm is motivated by the fact that GL functional converges to the TV seminorm [183, 313]; thus, 3.16 is modified using a diffuse interface approximation

$$E(u, c_r) = \epsilon \langle u, L_s u \rangle + \frac{1}{\epsilon} \sum_i W(u_i) + \mu \sum_{r=1}^{\hat{n}} \left\langle \|f - c_r\|^2, u_{\star,r} \right\rangle$$

Similarly to the procedure in the previous section (supervised), using gradient descent yields

$$\frac{\partial u}{\partial t} = -\epsilon L_s u - \frac{1}{\epsilon} W'(u) - \mu \left(\|f - c_1^n\|^2, \ldots, \|f - c_{\hat{n}}^n\|^2 \right)$$

One can use the MBO scheme, described in 3.3.1.2, to solve this minimization problem. For initialization of u in the unsupervised algorithm, random labels are used. To obtain the next iterate of u, one proceeds with following three steps:

Step 1 : Compute

$$\frac{u^{n+\frac{1}{2}} - u^n}{dt} = -L_s u^n - \mu\left(\|f - c_1^n\|^2, \ldots, \|f - c_{\hat{n}}^n\|^2\right)$$

Step 2 : Threshold

$$u_i^{n+1} = e_r, r = \arg\max u_i^{n+\frac{1}{2}}$$

for all $i \in 1, 2, \ldots, N$, where e_r is the r^{th} standard basis in $\mathbb{R}^{\hat{n}}$.

Step 3 : Update c

$$c_r^{n+1} = \frac{\left\langle f, u_{\star,r}^{n+1}\right\rangle}{\left\langle 1, u_{\star,r}^{n+1}\right\rangle}$$

The stopping criteria for this scheme is the same as the one in the previous section. The final classification of nodes is also obtained in the same manner as in previous section.

As in the case of the semisupervised algorithm. Step 1 can be computed very efficiently and simply by using the eigendecomposition of L_S. Let X be the matrix containing the first $m << N$ orthogonal leading eigenvectors of L, Λ be the diagonal matrix containing the corresponding eigenvalues, and write u^n as $u^n = Xa^n$. Then Step 1 of the algorithm can be approximately computed as

$$u^{n+\frac{1}{2}} = X(1 - dt \cdot \Lambda)a^n - dt \cdot \mu\left(\left\|f - c_1^k\right\|^2, \ldots, \left\|f - c_{\hat{n}}^k\right\|^2\right)$$

Due to the fact that, in practice, only the leading eigenvalues and eigenvectors need to be computed to obtain a good accuracy, 3.3.1.2 is an efficient way to compute Step 1 of the algorithm, even when the number of classes is large. This feature makes this method very fast. The algorithm also converges after a small number of iterations empirically. The iterations stop when a purity score between the partitions from two consecutive iterations is grater than 99.99%. In [212] Bertozzi et al. presented several applications of these techniques to image segmentation and data clustering and classification, showing how methods coming from image processing can be extended to different kind of data thanks to the graph setting.

3.3.2 Total Variation Scheme

Many clustering models rely on the minimization of an energy over possible partitions of the data set. These discrete optimizations usually pose NP-hard problems, however. A natural resolution of this issue involves relaxing the discrete minimization space into a continuous one to obtain an easier minimization procedure. Many current algorithms, such as spectral clustering methods, follow this relaxation approach. A fundamental problem arises when using this approach, however; in general the solution of the relaxed continuous problem and that of the discrete NP-hard problem can differ substantially. In other words, the relaxation is too loose. A tight relaxation has a solution that closely matches the solution of the original discrete NP-hard problem. Recently, a new set of algorithm that can obtain tighter relaxations than those used by spectral clustering provide a promising set of clustering tools. These all rely on the concept of total variation. In fact, total variation techniques promote the formation of sharp indicator functions in the continuous relaxation. These functions equal to one on a subset of the graph, zero elsewhere and exhibit a non-smooth jump between two regions. In contrast to relaxations employed by spectral clustering, total variation techniques therefore lead to quasi-discrete solutions that closely resemble the discrete solution of the original NP-hard problem. These techniques are broadly referred as "Total Variation Clustering" and were introduced in [302]. In the following we will describe the main concepts. Over the past several years, spectral clustering methods have become very popular; see [292] and [319] for an introduction. These methods start with a (nonnegative, symmetric) matrix W which collects the relative similarities between a set of points V to be clustered, and then makes the assumption that in some sense, the cluster indicators should be smooth with respect to W. A simple such notion is that the length of the boundary of the clusters should be small relative to their area. This motivates the definition of the Cheeger cut value of a partition $P = \{V_1, V_2\}$ of V into two pieces given by

$$\mathcal{C}(V_1, V_2) = \frac{Cut(V_1, V_2)}{\min(|V_1|, |V_2|)}, \tag{3.17}$$

where

$$cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} W_{ij} \tag{3.18}$$

and $|V|$ is just the cardinality of V. Since finding the optimal Cheeger cut is NP-hard, the Cheeger cut is usually approximated by the second eigenvalue of the combinatorial Laplacian $D - W$, where $D_{ii} = \sum_j W_{ij}$, such that:

$$\frac{1}{2\max_i D_{ii}}\mathcal{C}^2 \leq \lambda_2 \leq 2\mathcal{C} \tag{3.19}$$

See [75] for the continuous version and [78] for the discrete one. In [184] the parametric max flow-min cut (a.k.a graph-cut) was used to minimize the biased ratio cut $\frac{Cut(V_1,V_2)}{|V_1|}$, but cannot be used to solve the unbiased ratio cut defined as $Rcut(V_1, V_2) = \frac{Cut(V_1,V_2)}{|V_1|} + \frac{cut(V_1,V_2)}{|V_2|}$ which is NP-hard.

Using the Raleigh quotient formula for the eigenvalue gives

$$\lambda_2 = arg \min_{f \in L^2(V)} H_2(f) = arg \min_{f \in L^2(V)} \frac{\sum ||\nabla f||^2}{||f - M(f)||_2^2}, \tag{3.20}$$

where for $p \geq 1, ||\nabla f||^p$ at i is given by

$$||\nabla f||^p(i) = \sum_j W_i j |f(i) - f(j)|^p, \tag{3.21}$$

and where $M(f)$ is the mean of f. The functional $\mathcal{H}_2$ measures smoothness. It has long been known that L^2 measures of smoothness are not as well suited for dealing with functions jumps as L^1 measures of smoothness; in image processing, see for example [271]. In [12] and in [53] it was shown that

$$\lim_{p \to 1} \min_f \frac{\sum_i ||\nabla f||^p(i)}{\min_c ||f - c||_p^p} = min_P C(P) \tag{3.22}$$

With this in mind one can relax the problem

$$\min_P C(P) \tag{3.23}$$

as follows: for any binary valued function $f = \chi_{V_1}, V_1 \subsetneq V$,

$$||f - m(f)||_1 = \begin{cases} |V_2| & \text{if } |V_1| \geq |V_2|, \\ |V_1| & \text{if } |V_1| \leq |V_2|, \end{cases}$$

where $m(f)$ is the median of f, and V_2 is the compliment in V of V_1. Then

$$\frac{\sum_i ||\nabla f_P||(i)}{||f_P - m(f_P)||_1} = 2\frac{\sum_{v_i \in V_1} \sum_{vj \in V_2} W_{ij}}{min(|V_1|, |V_2|)} = 2\mathcal{C}(V_1, V_2) \tag{3.24}$$

thus

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - m(f)||_1} \tag{3.25}$$

is a relaxation of the Cheeger cut problem,

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - m(f)||_1} \leq \min_P C(P) \tag{3.26}$$

In [302] Bresson et al. showed the inequality 3.26 is actually an equality, and that for any solution of the relaxed minimization there is a threshold γ so that the binary function

$$f_\gamma = \begin{cases} 1 & \text{if } f \geq \gamma, \\ |0 & \text{if } f \leq \gamma, \end{cases}$$

has the same energy as the minimum cut. A similar approach has been studied in the continuous setting by Strang in [299]. In [302], authors give algorithms for minimizing the ratio energy and provide some experiments on the quality of the clusterings given by the algorithms presented.

3.3.3 Mumford-Shah Scheme

In this section we present an ongoing research whose aim is to show how the methods described in section 2.1.2 can be extended from image processing to data clustering. Our investigation of graph based Mumford-Shah functionals is motivated by problems arising in machine learning. Given a point cloud in Euclidean space with (noisy) real-valued labels or a graph with labeled vertices, we investigate a model to denoise the labels while allowing for jumps (discontinuities) in label values. As with the classical Mumford-Shah functional, this allows one to identify the locations of sharp transitions of label values. Our primary focus is on graphs arising as neighbourhood graphs of point clouds in an Euclidean space, in dimension two or higher, where we can carry out rigorous analysis. However some of the functionals we study can be formulated purely in the setting of weighted graphs and may be useful in applications.

Our scheme is based on an adaptation of the Mumford and Shah functional 2.1.2.4 to point clouds and graphs. We relay on the work of Gobbino [140] and

Gobbino and Mora [141] who introduced a family of non-local models which approximate the functional.

General graph setting. Considering an undirected weighted graph with vertices $V = 1, \ldots, n$ and edge weights matrix $W = [w_{ij}]_{i,j=1,\ldots,n}$. Edge weights are considered to be nonnegative and symmetric. Let $f : V \to \mathbb{R}$ be the observed noisy labels. Let $\zeta : [0, \infty)] \to [0, \infty)$ be concave and such that $\zeta(0) = 0$, $0 < \zeta'(0) < \infty$. For $p > 1$ we define the *Graph Mumford-Shah* functional acting on $u : V \to \mathbb{R}$ as

$$\mathcal{GMS}_f(u) := \frac{\lambda}{n} \sum_{i=1}^{n} |u_i - f_i|^2 + \frac{1}{\varepsilon n^2} \sum_{i,j=1}^{n} \zeta\left(\frac{1}{\varepsilon} |u_i - u_j|^2\right) w_{ij}$$

Minimizing the functional allows one to find the sharp transitions in the data by detecting edges where $u_i - u_j$ is large compared to ϵ. That is the parameter ϵ sets the scale for what differences of the values are considered "large". We note that functional is non-convex.

Geometric graph setting We now consider the setting of point clouds and the random geometric graphs generated by them. The ability to measure the distance between vertices allow us to create a larger family of graph Mumford-Shah functionals. Let $V_n = x_1, \ldots, x_n$ be a set of point in $\mathbb{R}^d$. The points x_i are typically random samples of a measure describing the data distribution, but this interpretation is not essential in defining the functional. Given these points we define a graph by setting the edge weights to be $w_i, i = 0$ and for $i \neq j$

$$w_i j = \eta_\epsilon(|x_i - x_j|)$$

where η is nonnegative, nonincreasing function which decays to 0 faster than a specified algebraic rate. Let $f : V_n \to \mathbb{R}$ be the observed noisy labels and let ζ be as in the graph setting above. For $p \in [1, d)]$ and $q \in [0, p-1]$ we define the *Graph Mumford-Shah* functional acting on $u : V_n \to \mathbb{R}$ as

$$\begin{aligned} \mathcal{GMS}_{f,\varepsilon,n}(u) := \frac{\lambda}{n} \sum_{i=1}^{n} |u(x_i) - f_i|^2 + \\ \frac{1}{\varepsilon} \frac{1}{n^2} \sum_{i,j=1}^{n} \zeta\left(\varepsilon^{1-p+q} \frac{|u(x_i) - u(x_j)^p}{|x_i - x_j|^q}\right) \eta_\varepsilon(|x_i - x_j|) \end{aligned} \tag{3.27}$$

We note that taking $q = 0$ reduces this functional to one considered in the

pure graph setting. We study the asymptotics $\mathcal{GMS}_{\epsilon,n}(u)$ as $n \to \infty$ and $\epsilon \to 0$ and establish that its minimizers converge to the minimizers of a Mumford-Shah functional posed in continuum Euclidean domain. The result can be extended to stochastic lattice model.

Related works Regularizing and denoising functions given on graphs has been studied in variety of contexts in machine learning. Here we focus on regularizations which still allow for the jumps in the regularized function. In our approach we take inspiration from image processing where variational approaches have been widely used for image denoising and segmentation. Particularly relevant in the context of imaging are the works of Chan and Vese [72, 73], who proposed a piecewise constant simplification of the Mumford-Shah functional and have shown its effectiveness in image segmentation, and Rudin, Osher and Fatemi [271] who proposed a TV (total variation) based regularization for the image processing denoising. As showed in a previous section, Bertozzi et al. formulated the piecewise-constant Mumford functional on graphs. They also developed an efficient numerical approach to compute the minimizers and used it to study a (multi-class) classification problem [167].

Continuum Mumford-Shah functional and its nonlocal approximation. We recall the Mumford-Shah functional using the formulation in the space of the special functions of bounded variation. For background on spaces of (special) functions of bounded variation we refer the reader to the book [9]. For $u \in SBV(\Omega)$

$$MS_f(u) := \lambda \int_\Omega |u - f|^2 dx + \int_\Omega |\nabla u|^2 \, \mathrm{d}x + \mathcal{H}^{d-1}(S_u) \tag{3.28}$$

where $f \in L^\infty(\Omega)$ is the noisy image, ∇u is the absolutely continuous (in the measure sense, and with respect to the Lebesgue measure) part of the gradient Du (which is a measure) of the function u, S_u is the jump set of u and $\mathcal{H}^{d-1}$ is the $(d-1)$-dimensional Hausdorff measure. The first term of the functional ensures the closeness of the approximation u to the original image f while the next two terms reward the regularity of u. The idea is that natural images are piecewise smooth, but often do have jumps in intensity between different regions. Thus the terms of the functional reward the regularity of u, while still allowing jumps in the intensity. Thanks to the work of Ambrosio in [5] and to the lower-semicontinuity of MS_f with respect to the topology of the space $SBV(\Omega)$, the direct method of

calculus of variations ensures us that a minimum $u_0 \in SBV(\Omega)$ for the functional 3.28 is always attained. For the considerations we have in mind the fidelity term $\int_\Omega |u-f|^2 dx$ is quite straightforward to treat. Hence, for readability, we introduce the functional without it and focus on this functional

$$MS(u) = \int_\Omega |\nabla u|^2 \, \mathrm{d}x + \mathcal{H}^{d-1}(S_u) \tag{3.29}$$

As shown in [47] any functional of the form of 3.29 cannot be approximated in the sense of Γ-convergence by *local* integral functional of the type

$$\int_\Omega h_\epsilon(\nabla u) \, \mathrm{d}x$$

for $u \in W^{1,2}(\Omega)$. De Giorgi conjectured that the MS functional can be approximated by nonlocal functionals. The conjecture was prove by Gobbino in [141], who showed that 3.29 can be approximated by the functionals

$$G_\varepsilon(u) := \frac{1}{\varepsilon^{d+1}} \int_{\mathbb{R}^d \times \mathbb{R}^d} \arctan\left(\frac{|u(y)-u(x)|^2}{|y-x|}\right) e^{-\frac{|y-x|^2}{\varepsilon^2}} \, \mathrm{d}x \, \mathrm{d}y$$

defined for $u \in L^1_{loc}(\Omega)$. He shows that for appropriate dimensional constants θ, σ

$$\Gamma - \lim_{\varepsilon \to 0} G_\varepsilon = \theta \int_\Omega |\nabla u|^2 \, \mathrm{d}x + \sigma \mathcal{H}^{d-1}(S_u)$$

where the Γ-limit is considered with respect to L^1 topology. The work [141] has been the generalized in [140] to functionals defined in $SBV(\Omega)$ of the form

$$F(u) := \int_\Omega \varphi(|\nabla u(x)|)\mathrm{d}x + \int_{S_u} \psi\left(\left|u^+(x) - u^-(x)\right|\right) \mathrm{d}\mathcal{H}^{n-1}(x)$$

where $u^+(x)$ and $u^-(x)$ denote the so-called approximate $liminf$ and $limsup$ of u at point x:

$$u^+(x) = \sup\left\{t \in \mathbb{R} : \lim_{r \to 0+} \frac{1}{r^n} |\{y \in B(x,r) : u(y) > t\}| > 0\right\}$$

They show that for suitable φ the functional can be approximated in the Γ-convergence sense with the family of non-local functionals of the form

$$F_\varepsilon(u) := \int_{\mathbb{R}^d \times \mathbb{R}^d} \varphi_{|x-y|} \left(\frac{|u(x) - u(y)|}{|x-y|} \right) \eta_\varepsilon(x-y) \mathrm{d}x \, \mathrm{d}y$$

where $\{\eta_\epsilon\}_\epsilon$ is a family of functionals and $\{\eta_\epsilon\}_{\epsilon>0} \subset L^1(\Omega)$ is a kernel.

Point cloud MumfordShah functional The above non-local approximation to the Mumford-Shah functional can be adapted to the graph setting. We consider the setting of random geometric graphs formulated on random samples of a measure μ with density ρ, which describes the underlying data distribution. Consider an open, bounded set with Lipschitz boundary Ω. The density ρ is assumed to satisfy: $\rho \in C^1(\Omega) \cap C^0(\overline{\Omega})$ and

$$0 < c \leq \min_{x \in \overline{\Omega}} \rho(x) \leq \max_{x \in \overline{\Omega}} \rho(x) \leq < \infty \tag{3.30}$$

We consider $\zeta : [0, \infty) \to [0, \infty)$ such that

(A1) ζ is concave and differentiable at 0;

(A2) ζ is not decreasing;

(A3) $\zeta'(0) < \infty$ and

$$\Theta := \lim_{x \to \infty} \zeta(x) \tag{3.31}$$

We fix $p \geq 1, q \in [0, p)$ and we assume that the kernel $\eta : [0, \infty) \to [0, \infty)$ satisfies

(B1) η is nonincreasing L^1 function, non identically 0;

(B2) $0 < \int_0^\infty \left(t^d + t^{p-q+d-1} \right) \eta(t) \mathrm{d}t < \infty$

In the sequel, we always assume the functions η, ζ and ρ to satisfy the above assumption. Let $x_1 \ldots x_n \in \Omega$ a set of n i.i.d random points on Ω chosen according to the probability measure $\mu = \rho \, dx$. The empirical measure of the sample is defined by

$$\mu_n = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$$

Given a Borel measure σ on Ω, the space $L^p(\Omega, \sigma)$ is the space of equivalence classes of measurable functions $u : \Omega \to \mathbb{R}$ with $\int_\Omega |u|^p d\sigma$ finite. Notice that,

under this assumption on ρ, we have that $L^1(\Omega;\rho) = L^1(\Omega)$. For that reason we often write $u \in L^1(\Omega)$ in place of $u \in L^(\Omega : \rho)$.

The graph Mumford-Shah functional we denote the most attention to is the functional 3.27 without the fidelity term. Namely for a function $u \in L^1(\Omega;\mu_n)$ let

$$\mathcal{GMS}_{\varepsilon,n}(u) := \frac{1}{\varepsilon}\frac{1}{n^2}\sum_{i,j=1}^{n} \zeta\left(\varepsilon^{1-p+q}\frac{|u(x_i)-u(x_j)|^p}{|x_i-x_j|^q}\right)\eta_\varepsilon(|x_i-x_j|) \tag{3.32}$$

Here $\eta_\epsilon(s) = \epsilon^{-d}\eta(s/\epsilon)$.

Main results The main result (see Theorem 10) states that as $n \to \infty$ and $\epsilon_n \to 0$ the Graph Mumford-Shah functional 3.32 converges to the following continuum Mumford-Shah functional:

$$MS_{\eta,\zeta}(u;\rho) := \vartheta_\eta(p,q)\zeta'(0)\int_\Omega |\nabla u(x)|^p \rho(x)^2 \,\mathrm{d}x + \sigma_\eta \Theta \int_{S_u} \rho(y)^2 \,\mathrm{d}\mathcal{H}^{d-1}(y) \tag{3.33}$$

defined for all $u \in SBV^p(\Omega)$ and where Θ is defined by 3.31 and

$$\begin{cases} \vartheta_\eta(p,q) := 2\omega_{d-1}\dfrac{\Gamma(p/2+1/2)\Gamma(d/2+1/2)}{\Gamma(p/2+d/2)}\displaystyle\int_0^\infty t^{p-q+d-1}\eta(t)\mathrm{d}t \\ \sigma_\eta := 2\omega_{d-1}\displaystyle\int_0^\infty t^d\eta(t)\mathrm{d}t \end{cases}$$

The $\Gamma-$convergence is the TL^1 sense defined in [307]. We point out that assumption (B2) on η is the one that guarantees the finitess of $\sigma_n, \vartheta_n(p,q)$. With all these in mind we are able to show the validity of the following statement.

Theorem 10. *Let Ω be an open set and ρ be a probability density satisfying 3.30. Consider ζ, η satisfying the assumptions (A1)-(A3) and (B1)-(B2). Let $\{x_i\}_{i\in\mathbb{N}}$ be a sequence of i.i.d. random points chosen accordingly to the density ρ and $\{\epsilon_n\}_{n\in\mathbb{N}}$ be a sequence of positive number converging to 0 such that*

$$\lim_{n\to\infty}\frac{(\log(n))^{1/d}}{\varepsilon_n n^{1/d}} = 0 \quad \text{for } d \geq 2 \tag{3.34}$$

Then $\mathcal{GMS}_{\epsilon_n,n}$ defined in 3.32, Γ-converges to $MS_{\eta,\zeta}(\cdot;\rho)$, defined in 3.29.

in the TL^1 sense.

3.4 Conclusions

In recent years algorithms based on the graphical framework have gained in popularity as highly competitive for problems involving high dimensional data. The success of such methods is partly due to the many advantages offered by using a graph-based approach. First, graphical techniques not only provide useful information about the overall data structure, but also about the connections between pieces of data via edge weights that encode the degree of similarity between pairs of nodes. Second, the graphical structure provides a general way to incorporate diverse types of data, such as hyperspetral data, text data, LIDAR data, images, video, etc.

A standard technique for data classification used in the context of machine learning involves minimizing a general form of energy (or cost) functional composed by a regularization term and a fidelity term. In the case of unsupervised classification, the second term is replaced by a region homogeneity term. When choosing the regularization norm, it is important to conserve the sharp discontinuities that may appear in the boundaries between classes. The model we studied is based on ideas from image processing and go back to the celebrated Mumford and Shah variational model for image segmentation that is particularly successful in denoising and recovering domain edges.

Several authors have proposed techniques for classifying data via Mumford and Shah functional minimization. We proposed a new one based on the mathematical framework showed in Chapter 2 that can be efficiently implemented on HPC architectures. Given a point cloud in a Euclidean space with (noisy) real-valued labels or an undirected graph with labeled vertices, our model denoises the labels while allowing for jumps (discontinuities) in label values. We tested our methods on image data, but we plan to apply it on different types of data.

Chapter 4

Variational Restoration of Curvilinear Structures

4.1 Introduction

In this chapter we will show the state of art in variational detection and restoration of curvilinear structures. Even though these techniques are successful at segmenting and reconstructing vessels in medical images, they do not provide satisfying results on images of cracks, due to their complex background textures and their high level of noise. Moreover their execution takes too long for a real-world application scenario. For this reason, we adapted them to the second order model described in subsection 2.1.2.5 and provided a parallel numerical implementation.

We recall that a curvilinear structure is any object with a spatial dimension d that is strictly lower than the dimension n of the space in which it is embedded. For $d = 1$ and $n = 2$, such objects can be vessels in eye fundus imaging, cracks on walls etc. The preservation of these structures is challenging for various reasons. They are generally sparsely distributed within images, due to their low dimension. They are also often thin structures, with a thickness similar to the image resolution. In addition, they can have complex topology with high curvatures, tortuosity, junctions and bifurcations and are easily corrupted by noise. For these reasons, many existing image processing methods cannot efficiently discriminate them from noise and artifacts. For tackling these issues, a solution may consist of guiding the restoration process by information related to the location and geomet-

ric properties (size, orientation) of these objects. Several contributions have been devoted to compute this kind of information from curvilinear structures in nD images (where $n = 2$ or 3). Two representative examples of such contributions are RORPO [217] and Frangi vesselness [125].

4.2 Curvilinear Structure Detection

In order to make a thin structure easily detectable, two main families of approaches have been developed: **linear** and **non linear filters**.

4.2.1 Linear Filters

In general, thin structures represent areas of higher intensity in the image. If we consider a grayscale 3D image as a discrete function of $\mathbb{R}^3$ in $\mathbb{R}$ (or a 2D image as a discrete function of $\mathbb{R}^2$ in $\mathbb{R}$) thin structures appear as its local maxima. For this reason the detection of such structures can be performed by studying the differential properties of the image.

This first family of filters relies on **linear operators**, based on local, differential analysis of images. In particular, the analysis of second order derivatives of 3D images were proposed in [281, 205]. In these pioneering works, the eigenvectors of multiscale Hessian matrices and their associated eigenvalues are analysed to characterise blobs (3D), planar (2D) and curvilinear (1D) structures, as well as their scale and orientation. The measure proposed in [125] is often considered the current gold-standard. Several variants have been proposed since then, for instance in [189]. Alternatively, steerable filters [128] can be expressed, for similar purposes, in terms of a linear combination of basis filters. Anisotropic diffusion methods [253, 325] were also proposed to filter curvilinear structures. Of particular interest is the Coherence Enhancing Diffusion (CED) filter [326], designed to perform actual anisotropic diffusion by adapting the diffusion direction using structure tensors.

There are three main classes of **filters based on derivatives**: the **gradient operator**, the **Hessian operator** and the **structure tensor**. The last ones are both more efficient than the gradient operator, which gives information only on the contour orientation. Indeed the Hessian operator and the structure tensor can capture characteristics related to the object shape (for example tubes or plans),

to the absence of structure (isotropic structure or "blob") or to noise.

The Gradient Operator The gradient operator is the vector of the first derivatives. It is given in 2D by:

$$\nabla(\mathrm{I}) = \begin{bmatrix} \frac{\partial \mathrm{I}}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$

or in 3D by:

$$\nabla(\mathrm{I}) = \begin{bmatrix} \frac{\partial \mathrm{I}}{\partial x} \\ \frac{\partial I}{\partial y} \\ \frac{\partial \mathrm{I}}{\partial z} \end{bmatrix}$$

where $\mathrm{I} : \Omega \subset \mathbb{R}^2(\mathbb{R}^3) \to \mathbb{R}$ is a C^1 function.

The gradient vector is characterized by its modulus and direction. Its modulus represents the image intensity slope in a specific point and its direction gives the direction of this slope. A high modulus reflects a large variation in gray levels around this point. We can therefore deduce that we are in a transition zone between a light part and a dark part, that is on a contour. In this case the vector is perpendicular to the contour. In practice, the gradient can be obtained from Sobel or Prewitt type filters, which calculate differences between neighbours in a given direction after smoothing. For example in 2D, the image is convolved with the following masks (the operator of Prewitt is defined for $k = 1$ and that of Sobel for $k = 2$)

$$\mathrm{G}_x = \begin{bmatrix} -1 & 0 & 1 \\ -k & 0 & k \\ -1 & 0 & 1 \end{bmatrix} \quad \mathrm{G}_y = \begin{bmatrix} -1 & -k & -1 \\ 0 & 0 & 0 \\ 1 & k & 1 \end{bmatrix}$$

The Roberts filter is the classical gradient operator consisting of the two masks:

$$\mathrm{G}_x = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \quad \mathrm{G}_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements

of the gradient component in each orientation (call these G_x and G_y). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of the gradient. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

The Roberts filter produces finer, more localized contours than other operators because it is the local differential filter of minimal size. Nevertheless, the filters of Prewitt and Sobel are more resistant to noise thanks to the low-pass filter.

The Hessian Operator The Hessian operator is the matrix of second derivatives. In 2D, the Hessian matrix of an image can be calculated in any pixel as

$$\mathcal{H}(\mathrm{I}) = \begin{bmatrix} \frac{\partial^2 \mathrm{I}}{\partial x^2} & \frac{\partial^2 \mathrm{I}}{\partial x \partial y} \\ \frac{\partial^2 \mathrm{I}}{\partial y \partial x} & \frac{\partial^2 \mathrm{I}}{\partial y^2} \end{bmatrix}$$

In 3D, the Hessian matrix:

$$\mathcal{H}(\mathrm{I}) = \begin{bmatrix} \frac{\partial^2 \mathrm{I}}{\partial x^2} & \frac{\partial^2 \mathrm{I}}{\partial x \partial y} & \frac{\partial^2 \mathrm{I}}{\partial x \partial z} \\ \frac{\partial^2 \mathrm{I}}{\partial y \partial x} & \frac{\partial^2 \mathrm{I}}{\partial y^2} & \frac{\partial^2 \mathrm{I}}{\partial y \partial z} \\ \frac{\partial^2 \mathrm{I}}{\partial z \partial x} & \frac{\partial^2 \mathrm{I}}{\partial z \partial y} & \frac{\partial^2 \mathrm{I}}{\partial z^2} \end{bmatrix}$$

In order to detect vessels of different sizes, it is important to apply these operators in a multi-scale setting. For that reason, one uses the linear theory of the "scale-space" which redefines the first and second derivatives as the product of convolution with the derivatives of the Gaussian function [182]

$$\begin{aligned} \frac{\partial}{\partial x}\mathrm{I}(\mathbf{x}, \sigma) &= \sigma^{\gamma}\mathrm{I}(\mathbf{x}) * \frac{\partial}{\partial x}\mathrm{G}(\mathbf{x}, \sigma) \\ \frac{\partial^2}{\partial^2 x}\mathrm{I}(\mathbf{x}, \sigma) &= \sigma^{\gamma}\mathrm{I}(\mathbf{x}) * \frac{\partial^2}{\partial x^2}\mathrm{G}(\mathbf{x}, \sigma) \\ \frac{\partial^2}{\partial x \partial y}\mathrm{I}(\mathbf{x}, \sigma) &= \sigma^{\gamma}\mathrm{I}(\mathbf{x}) * \frac{\partial^2}{\partial x \partial y}\mathrm{G}(\mathbf{x}, \sigma) \end{aligned} \tag{4.1}$$

where $G(x, \sigma)$ is the Gaussian function defined by

$$\mathrm{G}(\mathbf{x}, \sigma) = \frac{1}{(2\pi\sigma^2)^{\mathrm{D}/2}} e^{-\frac{|x|^2}{2\sigma^2}}$$

The parameter γ [201] to defines a normalization with respect to the σ scale. The larger the σ scale, the more noise in the image is removed while the contours are smoothed.

The Hessian matrix being symmetrical, it is diagonalisable. Its eigenvalues λ_1, λ_2 and λ_3 ($|\lambda_1| \leq |\lambda_2 \leq |\lambda_3|$) are respectively associated with eigenvectors $\mathbf{e_1}$,$\mathbf{e_2}$ and $\mathbf{e_3}$ forming an orthonormal basis. According to the theory of singular value decomposition, the vectors $\mathbf{e_1}$,$\mathbf{e_2}$ and $\mathbf{e_3}$ are oriented in the main directions of the tensor associated with the Hessian matrix. From the three directions and the three eigenvalues, we can then construct an ellipsoid representing the tensor described by the Hessian matrix or in other words the structure of the image at this point.

The analysis of orders of magnitude of these eigenvalues thus makes it possible to determine the shape (tubular, planar or isotropic structure) and the intensity (light, dark) of the local structures. For example, for an ideal tubular structure in a 2D image we have:

$$|\lambda_1| \approx 0$$
$$|\lambda_1| \ll |\lambda_2|$$

in a 3D image:

$$|\lambda_2| \approx |\lambda_3|$$

The Structure Tensor The structure tensor $g(\mathrm{I})$ is the matrix of the second order moments of the image gradient. In 2D. it is defined by:

$$g(\mathrm{I}) = \begin{bmatrix} \left(\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x}\right)^2 & \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y} \\ \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x} & \left(\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y}\right)^2 \end{bmatrix}$$

In 3D, it is defined by:

$$g(\mathrm{I}) = \begin{bmatrix} \left(\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x}\right)^2 & \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y} & \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial z} \\ \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x} & \left(\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y}\right)^2 & \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial z} \\ \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial z}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial x} & \frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial z}\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial y} & \left(\frac{\partial \mathrm{G}_\sigma(\mathrm{I})}{\partial z}\right)^2 \end{bmatrix}$$

where $G_\sigma(I)(x) = I(x) * G(x, \sigma)$ with $G(x, \sigma)$ is the Gaussian function previously defined. The tensor can be reformulated as the matrix product:

$$g(\mathrm{I}) = \nabla\left(\mathrm{G}_\sigma(\mathrm{I})\right).\nabla\left(\mathrm{G}_\sigma(\mathrm{I})\right)^{\mathrm{T}}$$

The structure tensor contains the principal directions of the gradient in the neighbourhood of a pixel x. It can be used in the same way as for the Hessian operator, although its interpretation is different. Since the tensor is semi-definite positive, all its eigenvalues are positive and can be ordered as before in 3D:

$$0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$$

In this order, λ_3 is the largest eigenvalue and its associated eigenvector $\mathbf{e_3}$ provides the principal direction of the gradient. If $\lambda_3 \gg \lambda_1 \approx \lambda_2$, then this means that the isosurfaces of I are planes perpendicular to $\mathbf{e_3}$. If $\lambda_3 \approx \lambda_2 \gg \lambda_1$, then the isosurfaces of I are tubes perpendicular to $\mathbf{e_3}$. If the three eigenvalues are of the same order of magnitude, the isosurfaces are isotropic structures. Identical to the case of the Hessian operator, we can reduce ourselves to a multi-scale framework by varying the parameter σ. On the other hand, the eigenvalues of the structure tensor do not make it possible to determine if the structures are clear or dark.

4.2.2 Non Linear Filters

Non-linear approaches include those based on mathematical morphology [230]. In this framework, a common notion is the structuring element (SE), a geometric pattern from which basic operators (erosions, dilations, openings, closings, etc) can be defined. The most popular approach uses a small and straight line of arbitrary orientation to carry out opening or closing operations [295], however the hard geometry remains a limitation to its accuracy and requires more flexible SEs [52]. Another approach is based on the more global notion of connectivity. In this context, the concept of component-tree [279] and attribute-based methods

were specifically investigated for threshold-based filtering. One example is area opening, which removes all the connected components whose area (number of pixels) is less than some threshold. This will, in general, not give the desired result, since short compact features will be preserved as well as the long, thin structures of interest. Another attribute that has been proposed for this problem is elongation. However, this one is not increasing [223], as required to form an opening and it becomes problematic if there are crossings in the image. The SE- and connectivity-based approaches present dual intrinsic strengths and weaknesses. SE-based approaches can naturally handle anisotropy, which is highly desirable for curvilinear structure filtering, but require explicitly defined families of SEs for orientation sampling. In comparison, connectivity-based approaches lead to more global descriptors, unfortunately the isotropic notion of adjacency cannot efficiently model the anisotropy of curvilinear structures. In [318], a notion of local optimal path was pioneered. The purpose was to restrict the search to a given distance, and in a given cone of orientation, in order to find the best paths starting from a given point. This paradigm led to the development of a notion of path operator [161], which uses a family of paths instead of a fixed shape as SE in order to enable a higher flexibility in geometry and size, while preserving a 1D semantics. The majority of oriented filters compute the response of a structure in an isotropic neighbourhood whose size depends on the scale, which itself depends on the size of the sought structure. This approach is not optimal for curvilinear structures which are highly anisotropic. In particular, it may lead to false detections and wrong orientation estimation, especially near structures borders. Path operators, by computing the response along an anisotropic neighbourhood fitting in the curvilinear structure, avoid this pitfall, moreover they are parameter-free: the only real parameter is the path length which is semantically related to the length of the structure of interest. The path operator is a kind of opening that removes bright features in which a path of length L does not fit. Since its introduction by Buckley and Talbot [161], several algorithms have been proposed to implement it. The one by van de Gronde [312] is the fastest, but it has the same time complexity as Appleton and Talbot's one [304]. It runs in $O(\min(L, d, Q)N)$ time, where L is the chosen length threshold, d is the maximum path length, Q is the number of gray levels and N is the size of the image. Moreover, it suffers from over-estimating the length because it tends to zig-zag in structures wider than one pixel.

Graph Path Operator In order to avoid this problem and to significantly increase the execution speed we propose an algorithm based on graph path opening. First of all we propose to preselect paths by an upper skeletonization [317]. This technique sets to zero some pixels, but preserves bright ridges. The algorithm goes on building a directed, acyclic graph from this preprocessed image: each non-zero pixel is represented as a node and each node is connected to its neighbours according to an adjacency graph G (traditionally S-N, SE-NW, E-W, NE-SW see figure 4.1).

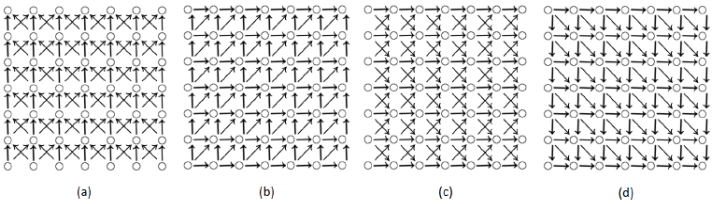


Figure 4.1: The four adjacency graphs for the path opening algorithm. Graph (a) shows the S-N adjacency graph, graph (b) shows the SW-NE graph, (c) is the W-E adjacency, and (d) is the NW-SE adjacency graph

Each of these graphs define adjacency relations between pixels in the image as follows: let $I \subset \mathbb{Z}^2$ and $x, y \in I$ are adjacent if (x, y) forms a directed edge in the adjacency graph G. It is easy to show that these ones are directed, acyclic graphs. The weights are chosen in order to minimize the relative error (i.e. the difference between the measured length and the actual length divided by the actual length) for digital lines of arbitrary orientation [259]: edges connecting diagonal neighbours are weighted by 1.340 and edges connecting horizontal/vertical neighbours are weighted by 0.948. Since paths are line-like, they will also minimize the error of paths reasonably well, assuming that they do not zig-zag inside thicker, line-like structures. Since the skeleton produces one-pixel thin preselected paths, the assumption holds. There are, however, other weights that could be used, depending on the error to be minimized (see [101]). After the graph has been generated a modified path opening is used to open this graph. For an adjacency graph $G = (V, E)$ and a set of image pixels I, let:

$$\nu_G(x) = \{y \in I : (x, y) \in E\} \tag{4.2}$$

be the set of all neighbours of x. Then, a tuple $a = (a_1, \ldots, a_L)$ is a path of

length L if $a_{k+1} \in \nu_G(a_k)$ for all $k \in \{1, 2, \dots L-1\}$. Let $\Pi_L^G(X)$ denote the set of all paths of length L in $X \subset I$ using the adjacency relation specified by G. For a path $a = (a_1, \dots, a_L)$, let $\sigma(a) = \{a_1, \dots, a_L\}$. Define the operator $\gamma_L^G(X)$ as

$$\gamma_L^G(X) = \cup_{a \in \Pi_L^G(X)} \sigma(a). \tag{4.3}$$

It is easy to check that the union of the elements of all paths of length L in X according to the adjacency graph G is an binary opening. In order to get the analogous operator for a grayscale image, one may decompose it in a series of binary image using thresholding, apply the binary path opening operator on each of these and finally combine the resulting openings into a single image, which corresponds to the grayscale opening. The only difference between the traditional algorithm and graph adaptation is that, instead of counting pixels, lengths are calculated from the edge weights that have been set during the graph creation. The main advantage is that, being easily implemented on High Performance Computing architectures, this version decreases in significant way the execution time.

4.2.3 Vesselness

Lorenz [205], Sato [282], and Frangi [125] used a tubularity measure or vesselness to detect tubular structures. All these measures calculate from Hessian matrix eigenvalues a criterion, which reaches strong values for a tubular structure and weak values elsewhere.

For a 3D image $I(x)$, observed at the σ scale, Lorenz proposes:

$$L(x) = \frac{|\lambda_1| + |\lambda_2|}{2|\lambda_3|}$$

which takes values close to 1 if you are in a tube, and values close to 0 otherwise.

Similarly, Sato proposes the following measure

$$S(x, \sigma) = \begin{cases} |\lambda_3| \left(\frac{\lambda_2}{\lambda_3}\right)^{\xi} \left(1 + \frac{\lambda_1}{|\lambda_2|}\right)^{\tau} & \text{if } \lambda_3 < \lambda_2 < \lambda_1 < 0 \\ |\lambda_3| \left(\frac{\lambda_2}{\lambda_3}\right)^{\xi} \left(1 - \rho\frac{\lambda_1}{|\lambda_2|}\right)^{\top} & \text{if } \lambda_3 < \lambda_2 < 0 < \lambda_1 < \frac{\|\lambda_2\|}{\rho} \\ 0 & \text{otherwise} \end{cases}$$

where ξ influences the asymmetry of the cross sections; $\tau > 0$ controls the sensitivity to the isotropic structure; and $0 < \rho \leq 1$ controls sensitivity to tubular structures.

For a 3D image, the vesselness of Frangi is

$$\mathcal{V}(x,\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{or } \lambda_3 > 0 \\ \left(1 - e^{\frac{-\mathrm{R}_\mathrm{A}^2}{2\alpha^2}}\right) \cdot e^{\frac{-\mathrm{R}_\mathrm{B}^2}{2\beta^2}} \cdot \left(1 - e^{\frac{-\mathrm{s}^2}{2c^2}}\right) & \text{otherwise} \end{cases}$$

with

$$\begin{aligned} \mathrm{R_A} &= \left|\frac{\lambda_2}{\lambda_3}\right| \\ \mathrm{R_B} &= \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}} \\ \mathrm{S} &= \|\mathcal{H}(\mathrm{I})_\sigma\| = \sqrt{\Sigma_j \lambda_j^2} \end{aligned}$$

The first term R_A discriminates the tubular structures from the planes, because it is close to 1 for a tube, while it is close to 0 for a plane. The second term R_B discriminates the isotropic structures from the tubes, because it is close to 1 for a blob, whereas it is close to 0 for a tube. Finally, S evaluates the noise level in a neighbourhood. The parameters α, β and c respectively control the filter sensitivity to R_A, R_B and S. For a 2D image, the vesselness of Frangi can be reformulated by

$$\mathcal{V}(x,\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \\ e^{\frac{-\mathrm{R}_\mathrm{B}^2}{2\beta^2}} \cdot \left(1 - e^{\frac{-\mathrm{s}^2}{2c^2}}\right) & \text{otherwise} \end{cases}$$

where $R_B = \frac{|\lambda_1|}{|\lambda_2|}$ distinguishes the isotropic structures from the thin ones. These models also admit a scale parameter σ used when calculating the first and second derivatives from the Gaussian function 4.1. Since this parameter is a priori unknown, a solution proposed by Sato and Frangi consists in calculating the response at each point of the image, $S(x,\sigma)$ or $V(x,\sigma)$ for several values of σ and take the maximum response. For example, for the Frangi's vesselness, we have

$$\mathcal{V}(x) = \max_{\sigma_{\min} \leq \sigma \leq \sigma_{\max}} \mathcal{V}(x,\sigma)$$

where σ_{min} and σ_{max} are the minimum and maximum scales for which structures of interest are expected to be found.

In the next section, we will show two hybrid methods that include the Frangi vesselness in a variational framework (the first in the ROF or TV-L1 denoising model and the second in the Chan-Vese segmentation model)

4.2.4 Inclusion of vesselness in the variational model

In [216] Merveille et al. introduced some traditional techniques to adapt image restoration (denoising/segmentation) to thin structures. Even though they showed interesting results in blood vessel restoration, they are not suitable for fracture segmentation, which requires a second order variational model as we showed in Chapter 2.

As first instance the authors presented a hybrid model in which Rudin Osher Fatemi (ROF) and L1 Total Variation (TV-L1) variational denoising models weights the data fidelity term by a vesselness prior. Actually, they replaced the coefficient λ in equations 4.6 and 4.5 by:

$$\lambda = \alpha\lambda_{reg} + (1-\alpha)\mathcal{V}_0(x) \tag{4.4}$$

where $\alpha \in [0,1]$ is a free parameter which balances the regularization and the vesselness (i.e. $\alpha \approx 0$ enhances the vesselness whereas $\alpha \approx 1$ promotes the regularization). Even though this solution made it possible to raise vessels in an image, it did not provide a proper segmentation. For this reason the authors showed a second example including a vesselness feature in the Chan-Vese variational segmentation model. In the context of tubular structures, several segmentation techniques have already been proposed by adding a shape prior, for example superellipsesoids [309], B-splines wavelet [303], adaptive dictionaries [264], and elastic regularization [105], but they all may have disconnections because the restoration has been carried out outside the structure of interest. In order to overcome this problem, Merveille et al. described how to include both vesselness and vessel directions in the Chan-Vese model. The directions were extracted from the Hessian matrix (we recall that they are the eigenvectors associated to the smallest eigenvalues in absolute value) and were integrated in the computation of the directional gradient. In particular they compared the directions extracted by Frangi's vesselness with the RORPO operator [217], which is

a tubular object filter based on mathematical morphology.

Hybrid Total Variation Model The ROF model corresponds to the following minimization problem:

$$\min_x \int_\Omega |\nabla x| + \frac{\lambda}{2} \int_\Omega (x-f)^2 dx \tag{4.5}$$

where x is the denoised image, f is the observed image, Ω is the domain of the image, while the parameter λ is used to manage the weighting between regularization and fidelity. We recall also that when the L^2 fidelity term is replaced by the L^1 norm, we obtain the TV-L1 model:

$$\min_x \int_\Omega |\nabla x| + \lambda \int_\Omega |x-f| dx. \tag{4.6}$$

Total variation is not appropriate for regularizing thin structures since it penalizes contours. By including the vesselness the hybrid model preserves better these shapes because it regularizes more outside than inside a tubular object. The parameter λ, which was previously a constant value for the whole image, now depends on the tubularity at each point (see formula 4.4), nevertheless the problem remains convex and can be addressed with traditional optimization techniques. Let $F, G \in \Gamma^0(\mathbb{R}^n)$ and $K : \mathbb{R}^n \to \mathbb{R}^n$ a linear operator, then 4.5 and 4.6 can be formulated in a general way:

$$\min_{x \in \mathbb{R}^n} F(Kx) + G(x) \tag{4.7}$$

This minimization problem can be solved by convex optimization algorithms such as the proximal algorithms [81]. In this case the primal-dual was chosen because it does not need any differentiability. In order to apply it to ROF and TV-L1 model, the proximal operators $prox_{\sigma F^\star}$ and $prox_{\tau G}\tilde{x}$ have to be defined.

Regarding ROF model, the regularization term is $F(Kx) = |\nabla x|$, the data fidelity is $G_{ROF}(x) = \frac{\lambda}{2}||x-f||^2$ and the linear operator $K = \nabla$ implies a Lipschitz constant $L^2 = ||\nabla||^2 = ||div||^2 \leq 8$ ([67] Theorem 3.1). The proximal operator for the primal variable is given by:

$$x = \text{prox}_{\tau \text{G}_{\text{ROF}}} \tilde{x} = \arg\min_x \left\{ \frac{\|x-\tilde{x}\|^2}{2\tau} + \frac{\lambda}{2}\|x-f\|^2 \right\}$$

Solving this minimization problem is equivalent to solving the associated Euler-Lagrange equation:

$$\frac{1}{\tau}(x - \tilde{x}) + \lambda(x - f) = 0$$

then, the solution is given by:

$$x = \frac{\tilde{x} + \lambda\tau f}{1 + \lambda\tau}$$

Finally, the proximal operator for the dual variable is given by:

$$y = \text{prox}_{\sigma \text{F}^*}\, \tilde{y} = \arg\min_{y} \left\{ \frac{\|y - \tilde{y}\|^2}{2\sigma} + \text{F}^*(y) \right\}$$

The conjugate by duality of F, $F^\star$, can be calculated using Chambolle algorithm [67]:

$$\begin{aligned} \text{F}(\text{K}x) = \|\nabla x\|_{2,1} &= \sup_{\{\xi \text{ such that } |\xi_{i,j}| \leq 1\}} \langle \xi, \nabla x \rangle \\ &= \sup_{\{\xi \text{ such that } |\xi_{i,j}| \leq 1\}} \langle -\operatorname{div} \xi, x \rangle \\ &= \sup_{p \in \mathbb{R}^n} (\langle p, x \rangle - \iota_{\text{P}}(p)) \end{aligned}$$

where $P = \{p = -\operatorname{div} \xi : |\xi_{i,j}| \leq 1 \forall i, j\}$ and ι_P is the P indicator function defined by:

$$\iota_P = \begin{cases} 0 & \text{if } p \in \text{P} \\ +\infty & \text{if } p \notin \text{P} \end{cases}$$

$F^\star(y) = \iota_P(y)$ and since it is the indicator function of a convex set, the operator proximal is reduced to a Euclidean projection on the unit ball:

$$y = \frac{\tilde{y}}{\max(1, |\tilde{y})}$$

The primal-dual algorithm for solving the ROF model is described in algorithm 4

For the TV-L1 model, one can use the same proximal operator for the dual variable $\text{prox}_{\sigma \text{F}^*}$; only the proximal operator for the primal variable changes: $\text{prox}_{\tau \text{G}_{TV-L1}}$ with $G_{TV-L1} = \lambda||x - f||$. This proximal operator is given by the following minimization problem:

Algorithm 4: Primal-dual algorithm for solving ROF model

$\tau = \sigma = \frac{1}{L} = \frac{1}{\sqrt{8}}, \theta = 1;$
$x_0 = f, y_0 = 0$ and $\overline{x_0} = x_0$
while $k \geq 0$ **do**
  $\tilde{y}_{k+1} = y_k + \sigma \nabla \tilde{x}_k;$
  $y_{k+1} = \frac{\tilde{y}_{k+1}}{\max(1, |\tilde{y}_{k+1}|)};$
  $x_{k+1} = (1 + \tau\lambda)^{-1}(\tilde{x_k} + \tau\lambda f);$
  $\tilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k);$

$$x = \text{prox}_{\tau G_{\text{TV-L1}}} \tilde{x} = \arg\min_x \left\{ \frac{\|x - \tilde{x}\|^2}{2\tau} + \lambda ||x - f|| \right\}$$

with the associated Euler-Lagrange equation:

$$\frac{1}{\tau}(x - \tilde{x}) + \lambda \frac{x - f}{|x - f|} = 0$$

that is equivalent to the following schema:

$$x = \begin{cases} \tilde{x} - \tau\lambda & \text{if } \tilde{x} - f > \tau\lambda \\ \tilde{x} + \tau\lambda & \text{if } \tilde{x} - f < -\tau\lambda \\ f & \text{if } |\tilde{x} - f| \leq \tau\lambda \end{cases}$$

The primal-dual algorithm for solving the TV-L1 model is described in algorithm 5.

Algorithm 5: Primal-dual algorithm for solving TV-L1 model

$\tau = \sigma = \frac{1}{L} = \frac{1}{\sqrt{9}}, \theta = 1;$
$x_0 = f, y_0 = 0$ and $\overline{x_0} = x_0$
while $k \geq 0$ **do**
  $\tilde{y}_{k+1} = y_k + \sigma \nabla \tilde{x}_k;$
  $y_{k+1} = \frac{\tilde{y}_{k+1}}{\max(1, |\tilde{y}_{k+1}|)};$
  $\tilde{x}_{k+1} = x_k + \tau \operatorname{div} y_{k+1};$
  $x_{k+1} = \begin{cases} \tilde{x}_{k+1} - \tau\lambda & \text{if} \tilde{x}_{k+1} - f > \tau\lambda \\ \tilde{x}_{k+1} + \tau\lambda & \text{if } \tilde{x}_{k+1} - f < -\tau\lambda \\ f & \text{si } |\tilde{x}_{k+1} - f| \leq \tau\lambda \end{cases};$
  $\tilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k);$

Hybrid Chan-Vese Segmentation Model The Chan-Vese segmentation problem corresponds to the following minimization problem:

$$\min_{\Sigma,c_1,c_2} \lambda \operatorname{Per}(\Sigma) + \underbrace{\int_{\Sigma} (c_1 - f)^2 \, dx}_{\text{region of interest}} + \underbrace{\int_{\Omega\setminus\Sigma} (c_2 - f)^2 \, dx}_{\text{background}} \tag{4.8}$$

where c_1 is the mean intensity in the region of interest Σ and c_2 is the average intensity in the background $\Omega \setminus \Sigma$ and $Per(\Sigma)$ is the perimeter of Σ.

Even though it is not convex, the problem 4.8 is equivalent to the following convex minimization problem [**?**]:

$$\min_{0\leq u\leq 1} \lambda \int_{\Omega} |\nabla u| + \int_{\Omega} \left((c_1 - f(x))^2 - (c_2 - f(x))^2\right) u(x) dx \tag{4.9}$$

And then $\Sigma = \{x : u(x) \geq \mu\}$ for all $\mu \in]0, 1]$. By setting $TV_c(u) = \lambda|\nabla u| + 1_{[0,1]}(u)$ the total variation under constraints and $\mathrm{A}(x) = (c_1 - f(x))^2 - (c_2 - f(x))^2$ the data fidelity, one can solve the equation 4.9 by using a backward-forward algorithm:

$$u(x) = prox_{\gamma TV_c}(u(x) - \gamma \mathrm{A(x)})$$

where γ is the algorithm step. The backward-forward algorithm is combined by a gradient descent on the differentiable function $\mathrm{A}(x)$ and the calculation of the proximal operator of the non differentiable function $TV_c(u)$. The difficulty lies on calculating the $TV_c(u)$ proximal operator under constraints given by:

$$\operatorname{prox}_{\mathrm{TV}_c}(f) = \arg\min_{0\leq u\leq 1} \|u - f\|^2 + 2\lambda \mathrm{TV}_c(u)$$

The minimization of total variation under constraints can be solved using Fast Gradient Projection algorithm (**FGP**) [24]. As in the previous models including vesselness, the λ coefficient is replaced by a parameter $\lambda(x)$ which depends locally on vesselness (see formula 4.4).

The Total Variation proximal operator without any constraint can be calculated using algorithm by Chambolle [67]:

$$\begin{aligned} u_{k+1} &= f - \lambda \operatorname{div} p_k \\ p_{k+1} &= \mathrm{P}\left(p_k + \tau \nabla u_{k+1}\right) \end{aligned}$$

where $P(\xi) = \frac{\xi}{\max(1,|\xi|)}$ is the projection on the unit ball, τ is the algorithm step and **div** is the discrete divergence. In [24], this algorithm is adapted to Total Variation under constraint TV_C:

$$u_{k+1} = P_C(f - \lambda \operatorname{div}(p_k))$$
$$p_{k+1} = \mathrm{P}\left(p_k + \tau \nabla u_{k+1}\right)$$

where P_C is the projection on the convex set C (here $C = [0, 1]$ in order to satisfy the constraint $0 \leq u \leq 1$).

The **FGP** algorithm for solving the problem 4.9 is described in algorithm 6.

Algorithm 6: Algorithm FGP (Fast Gradient Projection)

$u_0 = f, p_0 = 0$and $\tau > 0$ (algorithm step)
while $k \geq 0$ **do**
    $u_{k+1} = P_C(f - \lambda \operatorname{div} p_k)$;
    $p_{k+1} = \mathrm{P}\left(p_k + \tau \nabla u_{k+1}\right)$;

Directional Gradient Model Unfortunately, all these models do not provide an acceptable restoration for tubular objects because, by preventing denoising inside the structure, they allow disconnections and loss of smaller vessels. For tackling this issue one should find a way to guide the regularization along the main direction of an anisotropic shape. The notion of directional gradient operator ∇_D can provide a solution because it is based on two tubular priors: vesselness $\mathcal{V}$ and vessel direction. Several filters can provide them, for instance the Frangi [125] and the RORPO filters [217]. To define the directional gradient Merveille et al. consider a $2D$ image of size $N \times N$ and the family of unity vectors

$$\mathbf{v_1} = (\vec{i} + \vec{j})/\|\vec{i} + \vec{j}\|, \mathbf{v_2} = \vec{i}, \mathbf{v_3} = (\vec{j} - \vec{i})/\|\vec{i} + \vec{j}\|, \mathbf{v_4} = \vec{j}$$

The directional gradient ∇_D is defined by:

$$(\nabla_{\mathrm{D}} u)_{i,j} = \mathrm{D}^1_{i,j} \left(\nabla u\right)^1_{i,j} \mathbf{v_1} + \ldots + \mathrm{D}^4_{i,j} \left(\nabla u\right)^4_{i,j} \mathbf{v_4}$$

with $D^q, 1 \leq q \leq 4$ the matrix defined by:

$$\mathrm{D}^q_{i,j} = d^q_{i,j} \mathcal{V}_{i,j} + (1 - \mathcal{V}_{i,j})$$

where $\mathcal{V}$ is the vesselness and $(d^q)_{q\in[1,4]}$ are the direction coefficients. Intuitively, if $\mathcal{V} = 0$, i.e. the pixel (i, j) does not belong to a tube, then

$$(\nabla_{\mathrm{D}} u)_{i,j} = (\nabla u)^1_{i,j}\,\mathbf{v_1} + \ldots + (\nabla u)^4_{i,j}\,\mathbf{v_4}$$

is an isotropic gradient. If $\mathcal{V} = 1$, i.e. the pixel (i, j) belongs to a tube, then

$$(\nabla_{\mathrm{D}} u)_{i,j} = \mathrm{d}^1_{i,j}\,(\nabla u)^1_{i,j}\,\mathbf{v_1} + \ldots + \mathrm{d}^4_{i,j}\,(\nabla u)^4_{i,j}\,\mathbf{v_4}$$

is a directional gradient. Optimizing this new gradient no longer causes the contours of tubular structures to be lost since only the ends are captured.

In the following a way to compute the direction coefficients $d^q (q \in [1, 4])$ from the vesselness direction feature is described. As we have recalled a Hessian matrix has three eigenvalues λ_1, λ_2 and λ_3 ($|\lambda_1| < |\lambda_2| < |\lambda_3|$) associated respectively with the eigenvectors $\mathbf{e_1}$, $\mathbf{e_2}$ and $\mathbf{e_3}$. The principal direction of the vessel is then given by $\mathbf{e_1}$, corresponding to the smallest eigenvalue λ_1 [125].

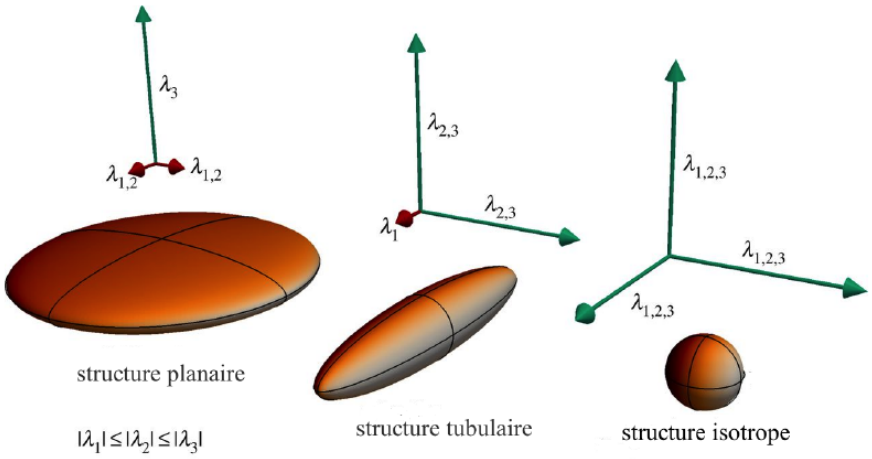


Figure 4.2: Ellipsoids representing the image structure.

Let us denote then $\delta_{\mathbf{i,j}} = (\mathbf{e_1}^x, \mathbf{e_1}^y)_{i,j}$ the vector giving the vessel local direction at pixel (i, j). The coefficients $(d^q)_{i,j}$, are then the coefficients of $\delta_{\mathbf{i,j}}$ decomposed on the family of vectors $(\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4})$. Note that if all the coefficients $(d^q)_{i,j}$ are different from 0, then ∇_D is similar to the isotropic gradient ∇. In order to obtain a proper gradient directional in 3D a way would be to impose that this decomposition would involve all zero coefficients, except for two vectors $\mathbf{v_m}$ and

$\mathbf{v_n}$:

$$\delta_{\mathbf{i,j}} = \sum_{q=1}^{4} d_{i,j}^{q} v_q = d_{i,j}^{m}\mathbf{v_m} + d_{i,j}^{n}\mathbf{v_n} \tag{4.10}$$

By multiplying equation 4.10 by $\mathbf{v_m}$, then by $\mathbf{v_n}$, one obtains the following system:

$$\begin{cases} d_{i,j}^{m}\mathbf{v_m}\cdot\mathbf{v_m} + d_{i,j}^{n}\mathbf{v_n}\cdot\mathbf{v_m} = \delta_{i,j}\cdot\mathbf{v_m} & \mathrm{L}_1 \\ d_{i,j}^{m}\mathbf{v_m}\cdot\mathbf{v_n} + d_{i,j}^{n}\mathbf{v_n}\cdot\mathbf{v_n} = \delta_{i,j}\cdot\mathbf{v_n} & \mathrm{L}_2 \end{cases}$$

By replacing $\mathbf{v_m}\cdot\mathbf{v_n} = \frac{\sqrt{2}}{2}$ because $(\mathbf{v_m},\mathbf{v_n}) = \frac{\pi}{4}$ and $\mathbf{v_m}\cdot\mathbf{v_m} = \mathbf{v_n}\cdot\mathbf{v_n} = 1$ because $\mathbf{v_m}$ and $\mathbf{v_n}$ are unit vectors, one gets:

$$\begin{cases} d_{i,j}^{m} + \frac{\sqrt{2}}{2} d_{i,j}^{n} = \delta_{\mathbf{i,j}}\cdot\mathbf{v_m} & \mathrm{L}_1 \\ \frac{\sqrt{2}}{2} d_{i,j}^{m} + d_{i,j}^{n} = \delta_{\mathbf{i,j}}\cdot\mathbf{v_n} & \mathrm{L}_2 \end{cases}$$

This is equivalent to

$$\begin{cases} d_{i,j}^{m} + \frac{\sqrt{2}}{2} d_{i,j}^{n} = \delta_{\mathbf{i,j}}\cdot\mathbf{v_m} & \mathrm{L}_1 \\ d_{i,j}^{n} - \frac{1}{2} d_{i,j}^{n} = \delta_{\mathbf{i,j}}\cdot\mathbf{v_n} - \frac{\sqrt{2}}{2}\delta_{\mathbf{i,j}}\cdot\mathbf{v_m} & \mathrm{L}_2 \leftarrow \mathrm{L}_2 - \frac{\sqrt{2}}{2}\,\mathrm{L}_1 \end{cases}$$

$$\begin{cases} d_{i,j}^{m} = -\frac{\sqrt{2}}{2} d_{i,j}^{n} + \delta_{\mathbf{i,j}}\cdot\mathbf{v_m} = -\sqrt{2}\delta_{\mathbf{i,j}}\cdot\mathbf{v_n} + 2\delta_{\mathbf{i,j}}\cdot\mathbf{v_m} \\ d_{i,j}^{n} = 2\delta_{\mathbf{i,j}}\cdot\mathbf{v_n} - \sqrt{2}\delta_{i,j}\cdot\mathbf{v_m} \end{cases}$$

$$\begin{cases} d_{i,j}^{m} = 2\delta_{\mathbf{i,j}}\cdot\mathbf{v_m} - \sqrt{2}\delta_{\mathbf{i,j}}\cdot\mathbf{v_n} \\ d_{i,j}^{n} = 2\delta_{\mathbf{i,j}}\cdot\mathbf{v_n} - \sqrt{2}\delta_{\mathbf{i,j}}\cdot\mathbf{v_m} \end{cases}$$

To calculate the proximal operator of the total variation under the constraint, one has to simply adapt the **FGP** algorithm to the directional gradient. Then, the first two iterations of algorithm 6 become:

$$\begin{aligned} u_{k+1} &= P_C(f - \lambda \operatorname{div}_D p_k) \\ p_{k+1} &= \mathrm{P}\left(p_k + \tau\nabla_D u_{k+1}\right) \end{aligned}$$

where $\nabla_D = D\nabla$ is the directional gradient with ∇ calculated as follows:

$$
\begin{aligned}
(\nabla u)^1_{i,j} &= u_{i-1,j-1} - u_{i,j} \quad \text{if } i,j > 0 \\
(\nabla u)^2_{i,j} &= u_{i-1,j} - u_{i,j} \quad \text{if } i > 0 \\
(\nabla u)^3_{i,j} &= u_{i-1,j+1} - u_{i,j} \quad \text{if } i > 0, j < \mathrm{N}-1 \\
(\nabla u)^4_{i,j} &= u_{i,j+1} - u_{i,j} \quad \text{if } j < \mathrm{N}-1 \\
(\nabla u)^n_{i,j} &= 0 \quad \forall n \in [1,4] \quad \text{otherwise}
\end{aligned}
$$

and div is the discrete divergence given by:

$$
(\operatorname{div} p)_{i,j} = p^1_{i,j} - p^1_{i+1,j+1} + p^2_{i,j} - p^2_{i+1,j} + p^3_{i,j} - p^3_{i+1,j-1} + p^4_{i,j} - p^4_{i,j-1}
$$

with

$$
p^1_{\mathrm{N}-1,\ \mathrm{N}-1} = p^2_{\mathrm{N}-1,j} = p^3_{\mathrm{N}-1,0} = p^4_{i,0} = 0 \forall i,j \in [0,\ \mathrm{N}-1]
$$

4.3 Conclusions

Classical variational restoration models such as ROF and TV-L1 are not appropriate to thin structures because they penalizes contours. For this reason Merveille et al. (see [216]) introduced a first model which includes a "tubularity measure" (i.e. Frangi's vesselness), which preserves these structures because it regularizes more outside rather than inside, but does not segment them properly. In fact, to obtain a segmentation a thresholding is necessary, but this process make darkest structures disappear.

The Chan-Vese model is a segmentation model that can detect regions of interest from background. Merveille et al. proposed a first model that includes only Frangi's vesselness, which outperforms ROF and TV-L1, but presents disconnections. In order to overcome this problem they introduced a model including both vesselness and structure direction. The latter one is extracted from vesselness and is integrated into the gradient, which is no longer a classic gradient, but a directional gradient. This model provides better results in terms of accuracy and structure reconnection (see [216]).

In order to segment only anisotropic structures like cracks, we apply this technique to the mathematical model described in 2.1.2.5. In particular, we tried two morphological filters (4.2.2,[217]) that showed equivalent results (see 5.2.2 in Chapter 5).

.

Chapter 5

Computer Vision Techniques for Inspection of Large Concrete Structures

5.1 Introduction

Infrastructures can be exposed to different loading conditions, recurrent ones due to vehicular traffic and extraordinary ones caused by earthquakes, wind, and strong rain. The consequently induced stresses may determine structural deterioration and damage, which can even cause catastrophic collapses [255]. Therefore, the issues related to the possibility of reaching/increasing a level of automation for inspection and maintenance of infrastructure are still under research. During these last years, the classical activities conducted mainly by human inspectors through visual quality control for damage assessment is under reconsideration due to newly available tools coming from information and communication technologies. For example, current visual inspection, which highly relies on an inspectors subjective, error-prone and empirical knowledge [180], can be enhanced by robotic/automatic assisted operations [260]. Usually, the actions performed by inspectors require a long time to examine large areas, which may be also difficult to access. Inspections often need to be performed with specialized equipment like large under bridge units, heavy trucks, special elevating platforms or scaffolding on structures. Altogether these solutions are in most cases expensive and cause high logistical efforts and costs. Moreover they can even interfere with the

operational conditions of structures and infrastructure.

Recent works address the problem of the automation of inspection and maintenance tasks based on robotic systems [199]. Existing automatic or robotic systems based on ground or aerial solutions have been proposed for inspection of dangerous sites or those difficult to access, but at the present state-of-the-art, human-based procedures are not yet completely substituted. Examples of ground systems used for inspection are wheeled robots [163] and legged robots [119]. In case of inspection of vertical surfaces, wall-climbing robots were developed using magnetic devices [157] or using vacuum suction techniques [283] and remote-controlled unmanned aerial vehicles (UAVs), equipped with high-definition photo and video cameras, were used to get high-quality data. In particular, (UAVs) have showed great potential not only in inspection applications [159, 272, 174], but also in additive building manufacturing [86].



Figure 5.1: Inspection with drones

Most of the infrastructure and civil structures are made by concrete, steel and masonry, which are prone to cracks due to creep, shrinkage and corrosion of reinforcements. Actually, crack information (e.g., the number of cracks and crack width and length) represents one of the current structural health indicators [204]. Nowadays damages in buildings and bridges can be easily captured using a commercial digital camera and consequently analysed and classified by image processing algorithms, but the detection of fractures is still challenging in

image processing. The main reasons are that they have a complex topology, a thickness similar to the image resolution and are easily corrupted by noise. The most frequently used techniques are those based on colour detection. In [179] a comparative analysis is proposed among different colour spaces to evaluate the performance of colour image segmentation using an automatic object image extraction technique. In [171] an RGB based image processing technique was proposed for rapid and automated crack detection. Even though these techniques allow fast processing and are highly robust to geometric variations of object patterns and viewing directions, they are not suitable for inspections because they are too sensitive to change in environmental conditions and noise. Recently, algorithms based on Convolutional Neural Networks have showed promising results. In particular, [64] uses these techniques to detect concrete cracks without calculating the defect features [65], [200]. Furthermore, a Fusion Convolutional Neural Network is proposed and employed in [338] for crack identification in steel box girders containing complicated disturbing background and handwriting. As stated in [261] these methods are affected by a high incidence of false alarms and need to be combined with pre/post processing techniques in order to process corrupted images.

Since many damages and many defects are characterized by their distinctive visual appearance (different shape, colour or texture), computer vision-based approaches can be used to automatically recognize, classify and measure them in image and video data.

5.1.1 Autonomous Robotic Systems for Defect Detection

5.1.1.1 San Giorgio's Bridge

The San Giorgio bridge, which was designed by the renowned architect Renzo Piano, replaced the Morandi Bridge that dramatically collapsed in August 2018. Setting up an ideal connection between the bridge and the essential maritime character of the city of Genoa, Piano designed the box steel girder supporting the deck with a unique elliptical shape that resembles the keel of a ship. The girder, whose components were built at different Italian shipyards, continuously spans for 1067 metres over 18 reinforced concrete piers. The continuity of the girder is allowed by an advanced bearing system that isolates the continuous girder from the piers and also protects the bridge against potential seismic activity. Between

the many outstanding qualities of Renzo Piano, there is his willingness to embrace the most advanced construction technologies. He has been known to speak of maintenance as an act of "care" toward bridges and buildings that ultimately make them last long. In this case he combined his focus on advanced technologies and his emphasis on maintenance by envisioning the use of mobile robots permanently installed on the bridge deck. The task of bringing Piano's vision to fruition was taken by the Industrial Robotic Unit (IRU) of Italian Institute of Technology (IIT). On the San Giorgio bridge are installed two robots, the RobotInspection and the RoboWash (see figure 5.2)



Figure 5.2: RoboWash and RobotInspection

Both the RobotInspection and the RoboWash move longitudinally along the rails placed at the two sides of the deck making a total of four robots. The two robots look distinctively different from one another.

The RobotInspection is essentially a carbon fiber beam with a fixed section and a retractable one. When the retractable beam is fully elongated, the RobotInspection reaches 17 meters in length. The robot weighs 2200 kg; it is 7-meters wide and is anchored to the rails with 56 wheels. Another 26 wheels are in place for moving the retractable beam. The robot moves at a rate of 100-150 mm/s over the rails. As reported in the bridge's inspection manual, the RobotInspection is responsible for fully autonomously monitoring the exterior of the steel girder. This is the robot's primacy compared to current robots in use. As an additional feature, this robot is also suitable for semi-autonomous inspection of the bearings. The fixed and retractable sections of the RobotInspection monitor

the outer surface of the steel box girder, which has an elliptical shape, by taking approximately 20000 pictures over the $30,000\ m^2$ of outer surface of the girder. The RobotInspection has the ability to take up to $25,000$ photos in a few days if weather and light permit.

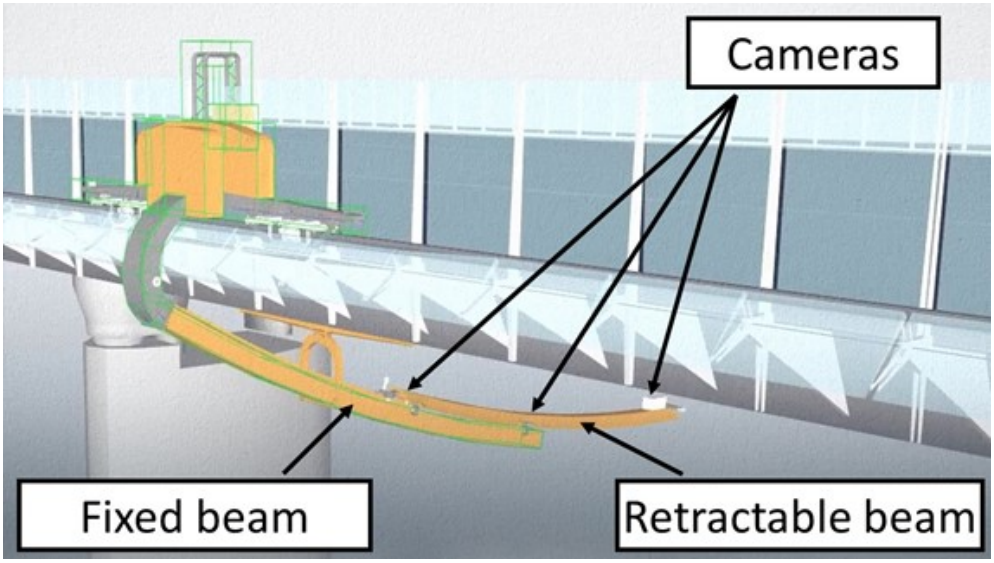


Figure 5.3: Inspection Bot: Full Extension of the Retractable Beam

The retractable beam is equipped with 3 cameras that have the capacity to scan the full outer surface of the steel girder. Scanning proceeds from the top level to the bottom level of the girder, which can only be reached when the retractable beam is fully extended. Essentially the RobotInspection works as a scanner taking photos of the outer surface of the steel box girder. Each photo covers a surface of approximately 1 m^2. By analysing the photos taken by the RobotInspection, bridge maintenance expert personnel can detect early signs of deterioration, such as paint flacking and/or steel corrosion. They can also examine the conditions of welding and connections. What sets this monitoring system apart from conventional inspection methods is the sheer amount of information collected and, even more importantly, the total objectivity of data. When photos are compared over time, there is total consistency of information due to the fact that photos are taken by the same equipment, at the same distance, and at the same angle. This level of data accuracy, consistency and repeatability cannot be achieved by drones or by inspections carried out by individuals, whose reporting always contains subjective elements of evaluation. Even if inspections are carried

out by the same individual, this individual cannot guarantee that two or more reports will not be somehow affected by his subjectivity. The RobotInspection can be equipped with an additional retractable beam that is connected to the retractable section of the main beam. The main beam is a huge structure that can carry up to 80 kg on its end. The additional beam has the ability of moving toward the surface of the steel girder to the point of touching it. It is designed to carry specialized instruments, such as $3D$ camera and ultrasound sensors that can provide in-depth information of steel imperfections. Ideally, in the future, the additional beam could also be equipped with instruments for painting and touching up. The additional beam is designed to be used ad hoc. For example, if pictures taken by the main beam show 3 or 4 anomalies in the girder's steel surface, the owner has the capability of using the additional beam to evaluate these anomalies. If one of these anomalies remains questionable after the second inspection, then it is time to send an inspector. As a result, the robot has reduced the use of inspectors to a bare minimum, thus lowering costs and risks. It is the bridge owner that decides on the frequency of use of the RobotInspection: it operates from one to two times per year. It takes a few days for the RobotInspection to complete the full inspection of the continuous girder. This time varies depending on weather conditions. The robot is equipped with sensors that stop its functioning in conditions of extreme weather, such as heavy rain or wind gusts of more than 15 meters/second. The RobotInspection also stops working if the light is not sufficient to take pictures. Having a memory, the robot resumes operating from where it stopped. The large number of $2D$ pictures taken by the RobotInspection are sent in real time to the data base of bridge's Control Center, which is equipped with a custom-designed software that contains algorithms for data analysis and storage. The RobotInspection is considered part of the monitoring system of the bridge, which also encompasses more than 240 sensors that are embedded in the bridge structure. These sensors include 70 inclinometers, 50 accelorometers, and 50 extensometers.

5.1.1.2 ROBO-SPECT Project

Tunnel's structural evaluation and maintenance is an important task in civil transportation infrastructures. It is a tedious and resource consuming operation, mainly performed through tunnel-wide visual observations by inspectors; a

human must identify structural defects, evaluate them and then, based on their severity, categorize them. Such human-involving approaches have serious drawbacks: they reduce the operational uptime of tunnels and are often incomplete due to fatigue, lack of experience, subjectivity and adverse working conditions. For these reasons the adoption of automated procedure can significantly improve the inspection quality.

The framework presented in [260, 261] has been deployed on board an integrated, autonomous robotic system for tunnel inspection, which was designed, implemented and validated in the context of the EU ROBO-SPECT project. The robotic inspector was tested in a tunnel of Egnatia Motorway in Metsovo, Greece. Experimental results were obtained using a i7 quad core processor, 8GB RAM and GPU available. The code was developed in Python. The constraints involved in such a real-world functional system setting make crack detection a far more challenging task compared to a standalone desktop application evaluating photographs of concrete tunnel surfaces from a dataset. The significant challenges are: limited processing resources on board the robotic platform; seamless integration with other components of the platform; difficult deployment conditions; need for increased detection accuracy at fast response times, since the presented computer vision framework drives the precision positioning of a sensitive ultrasound sensor around the crack to further evaluate the defect attributes; limited training samples; need for keeping the overall computational cost at reasonable levels.

Figure 5.4 displays the components of the robotic platform and Figure 5.5 shows the robotic platform in action.

The main goal of the computer vision module is to identify cracks over images acquired on the robotic camera and pinpoint the precise location in which a depth sensor would be inserted in order to further investigate the condition. The problem was limited to a traditional binary classification approach.

Visual inspection systems should ideally be able to identify cracks on an RGB image of a concrete surface as successfully as the human eye. Their performance should not be affected by angle and distance from the surface, nor from illumination conditions. Proximity sensors, advanced navigation and lighting equipment facilitate the acquisition process, but cannot guarantee ideal conditions or eliminate occlusions (e.g. by wiring).

However, even under ideal conditions, the variability of defect types make the

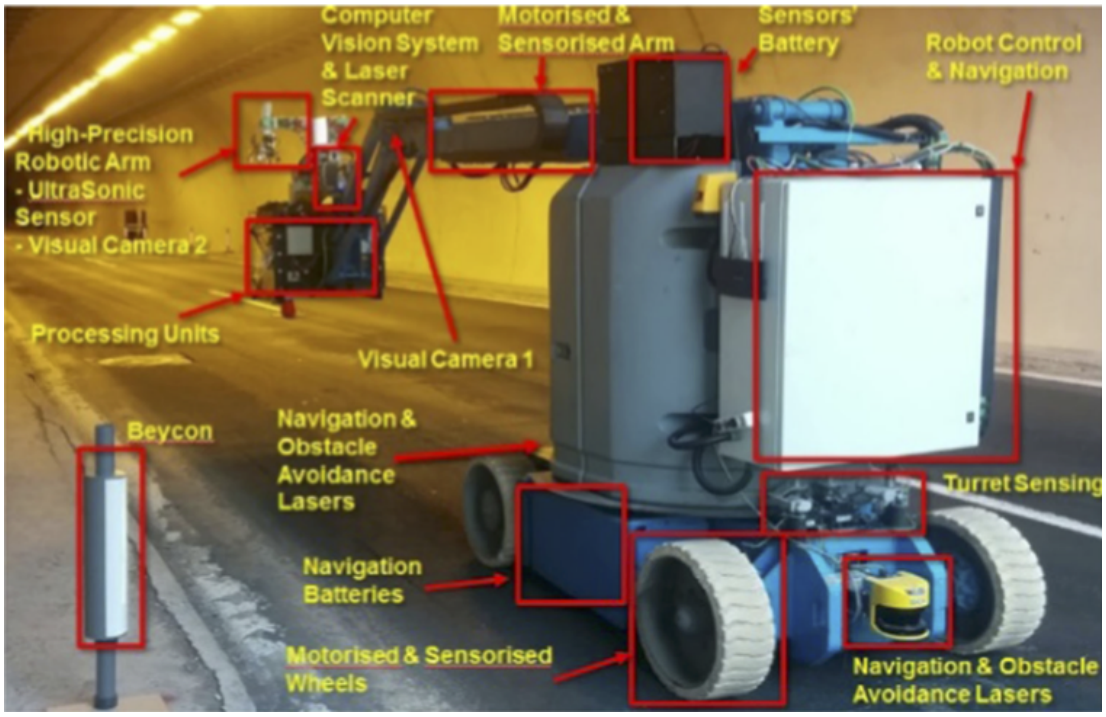


Figure 5.4: The components of the ROBO-SPECT robotic platform

problem increasingly difficult for a detection mechanism. There are various defect types, which makes feature construction/selection a cumbersome task. The term "defect" can be interpreted in many ways; deformations, cracks, surface disintegration, spalling and other defects are widely known and commonly appear. Other defects include discoloration of the concrete, small voids (bug holes) on the surface of vertical concrete placements and honeycombing, which is the presence of large voids in concrete. This variety in defect types hinders the feature extraction process making it difficult to construct appropriate descriptors for representation. Cracks appear on concrete surfaces usually as secondary symptoms of other defects. As such, the identification of a crack should be the first step, prior to an extensive analysis in the surrounding area, using laser scanners, ultrasounds or other approaches. Moreover, the variability of defect types, the existence of noise as well as other surface formations, which resemble cracks but are not and finally the scarcity of training data further underscore the challenging nature of the task at hand.

Crack detection is performed by a CNN-based classifier that annotates an RGB image (see for details [260]). Initially, an RGB image pair is captured.

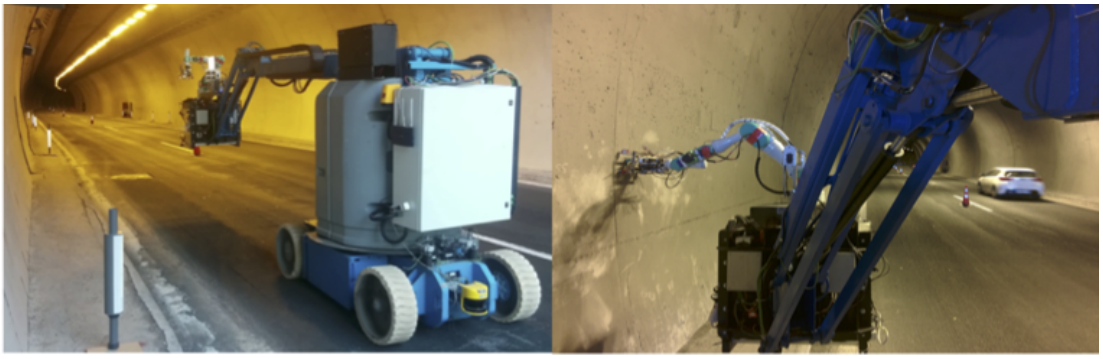


Figure 5.5: Left: Visual inspection of a tunnel at the Egnatia Motorway by the robotic paltform. Crack detection is driven by the presented computer vision component. Right: The robotic arm positioning the ultra-sonic sensor on a crack detected by the computer vision component



Figure 5.6: Illustration of a noisy case resulting in falsely identified cracked areas

The detection mechanism utilizes only one of these two images; the second one is exploited in case of a positive detection for $3D$ reconstruction. Grayscale and resize operators are applied to the image, prior to the assessment by the CNN detector. Both operators, as well as other techniques, were used in order to reduce detection times. Unfortunately, it is likely that unexpected occurrences appear on the tunnel surface. Given the scarcity of training data, it is very difficult for the CNN to train for all possible cases including outliers. Therefore, the CNN will probably handle a cracked area as anything that looks like a crack and was not modelled during training. Even though in [261] a post-processing heuristics-based mechanism is applied in order to avoid the false positives, we think that the variational methods for curvilinear structures restoration we showed in the section 5.2.2 could be even more useful in filtering out all the noise sources in the images. The main advantages are: 1) the procedure is fully automated; 2) it does not require any context knowledge; 3) it can be executed efficiently on huge amount of data on any commodity server; 4) it can be integrated in any Build

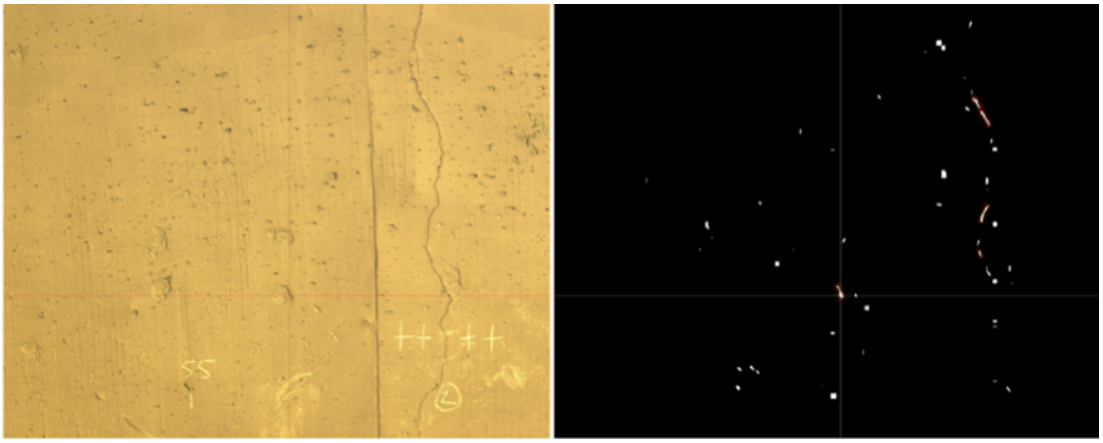


Figure 5.7: Illustration of wrong crack position indication. The CNN annotations have falsely marked as potentially cracked areas noisy regions of the image.

Management Information platform.

5.2 Crack Detection

5.2.1 Introduction

Many machine vision-based methods have been created to automatically detect cracks on structural concrete member surfaces. These methods are generally classified into three categories. The first category recognizes only whether or not an image contains a crack (crack presence). For example [2] propose a principal component analysis (PCA) based algorithm for recognizing crack presence in concrete bridge surface images. In their algorithm, an image is first segmented into square blocks. Each block is filtered by linear feature detectors (horizontal, vertical and oblique) and then projected onto dominant eigenvectors that are pre-generated using a training data set. The projection result is further compared with the projection results of training data to determine the presence of cracks in each block. This way, cracks in an image can be recognized sequentially on the basis of these blocks.

Methods that belong to the second category can also locate crack points in an image a produce a crack map. A crack map is binary image in which each isolated crack pixel is marked white, and non-crack pixels are marked black. These methods use special visual characteristics of cracks in images and adopt various image

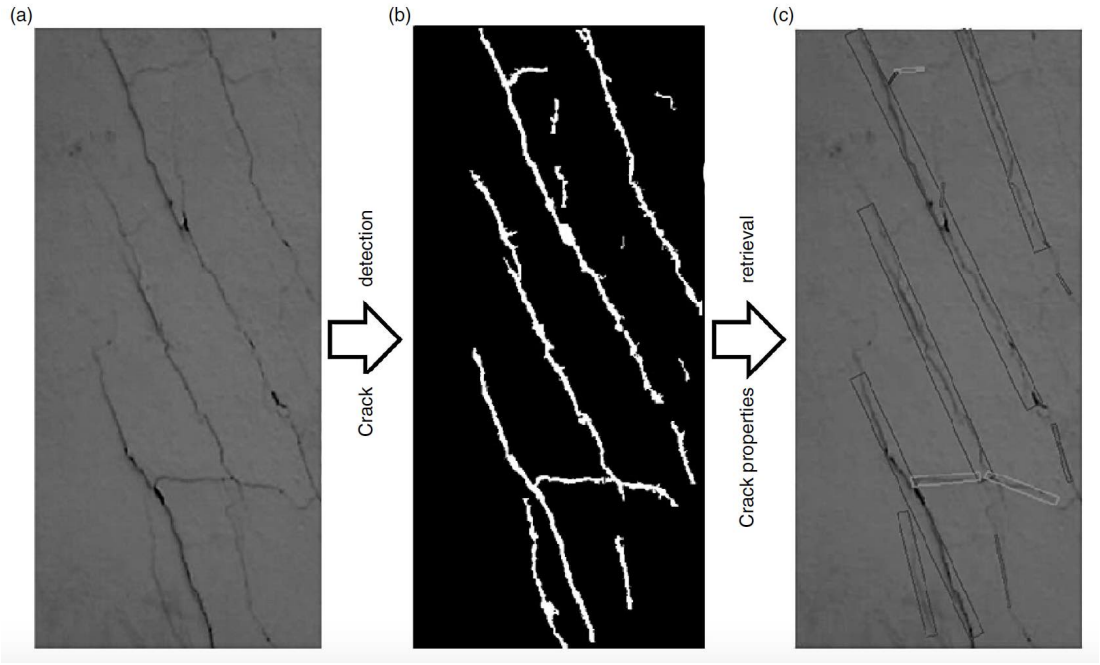


Figure 5.8: Crack detection and properties retrieval results: (a) original crack image; (b) crack map; and (c) crack segments

processing techniques, such wavelet transforms, threshold, and edge detection (Canny edge detector, Sobel edge detector, Fourier transform, fast Haar transform), to extract pixels from the image background. [77] detects cracks by simply thresholding the concrete surface image. The threshold value is determined based on the image's mean and standard deviation values. However, in these global processing methods detection accuracy is affected by image noise. To address the problem of image noise, [339] propose a fast and scalable local percolation-based image processing method that considers crack connectivity among neighbouring image pixels.

The third category contains methods that use crack maps to retrieve crack properties like length, maximum width, average width and orientation. [340] calculate the length, width and orientation of cracks through graph search; however, their method required the start and end points of the crack to be manually provided first. [66] use an artificial neural network to retrieve crack properties. [344] propose a method that creates topological skeletons of cracks through binary image thinning and calculates the distance field of crack pixels in the map using a distance transform. According to skeleton configurations and the distance

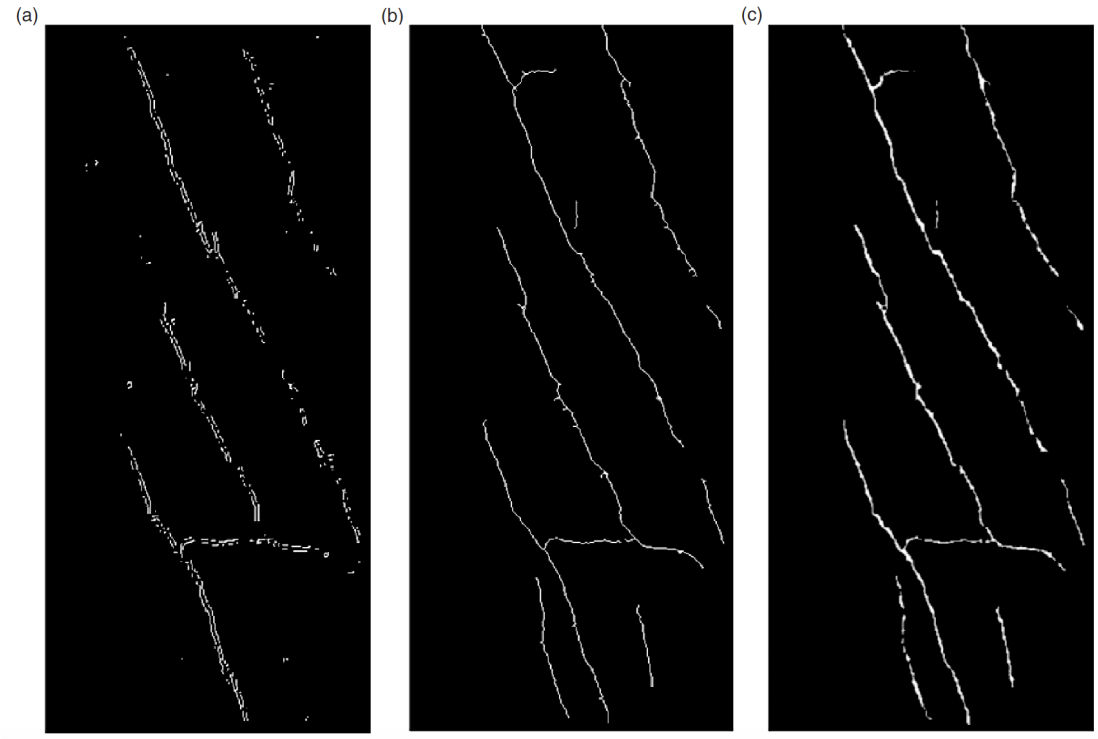


Figure 5.9: Crack property retrieval procedure and results: (a) edge map; (b) crack skeleton; and (c) distance map

values of crack pixels, crack properties (width, length, orientation and location) are retrieved with an average error of 3%.

5.2.2 Automatic Crack Detection with Calculus of Variations

Variational methods have successfully addressed problems such as image segmentation and edge detection. They propose as solution a minimizer of a global energy. A first example is described by Mumford and Shah (MS) (see section 2.1.2.4) where they proposed a first order functional, whose minimization determines an approximation of the image by means of a piecewise smooth function and detects edges as singularities in the image intensity. However, this model is not suitable for cracks because they do not represent singularities in the intensity function, but in its gradient instead. For this reason, we propose a second order variational model based on the Blake-Zissermann (BZ) functional see section 2.1.2.5). This was introduced with the aim of overcoming some limitations

of the MS approach, such as the over segmentation and the lack in detecting gradient discontinuities. With the original formulation being not suitable for numerical treatment, we had to work on a different approach that is based on the approximation proposed by Ambrosio and Tortorelli (AT) for the MS functional [8]. In their model, they replaced the unknown discontinuity set by an auxiliary function which smoothly approximates its indicator function. In our case two auxiliary functions are introduced as indicators of both intensity discontinuity and gradient discontinuity sets. The qualifying terms "free discontinuities", "free gradient discontinuities" mean that the functional is minimized over three variables: two unknown sets K_0, K_1 with $K_0 \cup K_1$ closed, and u, a smooth function on $\Omega(K_0 \cup K_1)$ as follows:

$$\begin{aligned}\mathcal{F}(u, K_0, K_1) = &\int_{\Omega\setminus(K_0\cup K_1)} (|\nabla^2 u|^2 + \Phi(x,u))dx \\ &+\alpha\mathcal{H}^{n-1}(K_0 \cap \Omega) + \beta\mathcal{H}^{n-1}((K_1\setminus K_0)\cap\Omega)\end{aligned} \tag{5.1}$$

α and β being two positive parameters. The set K_0 represents the set of jump points for u, and $K_1 \setminus K_0$ is the set of crease points of u, those points where u is continuous, but ∇u is not. Under certain conditions, the existence of minimizers for Blake-Zisserman functional is ensured over the space $\{u : \Omega \subset \mathbb{R}^n \to \mathbb{R} | u \in L^2(\Omega), u \in GSBV(\Omega), \nabla u \in (GSBV(\Omega))^n\}$, being $GSBV(\Omega)$ the space of generalized special functions of bounded variation. By properly adapting the techniques of [8], two auxiliary functions $s, z : \Omega \to [0,1]$ (aimed at approximating the indicator functions of the discontinuity sets) are introduced to the model and a $\Gamma-$convergence approximation of $\mathcal{F}$ is proposed via the following family of uniformly elliptic functionals

$$\begin{aligned}\mathcal{F}_\epsilon(u,s,z) =&\delta\int_\Omega z^2\left|\nabla^2 u\right|^2 dx + \xi_\epsilon\int_\Omega \left(s^2+o_\epsilon\right)|\nabla u|^2 dx + (\alpha-\beta)\int_\Omega \epsilon|\nabla s|^2 + \\ &\frac{1}{4\epsilon}(s-1)^2 dx + \beta\int_\Omega \epsilon|\nabla z|^2 + \frac{1}{4\epsilon}(z-1)^2 dx + \mu\int_\Omega |u-g|^2 dx\end{aligned}$$

where $(s,z,u) \in [W^{1,2}(\Omega,[0,1])]^2 \times W^{2,2}(\Omega) = \mathcal{D}(\Omega)$.

For the numerical minimization algorithm, we chose an inexact block-coordinate descent scheme (BCD) in order to address the heterogeneous hardware environ-

ment. Although the model is global, several numerical experiments have highlighted that the solutions weakly depend on boundary conditions. This motivates the adoption of a tiling scheme to address very large images: the minimizer is assembled by merging together local solutions restricted to portion of images.

Regarding the implementation, initial results pointed out the need to find an approach that increases data locality: this feature can be achieved by partitioning data and variables and considering independent subproblems. In this approach data dimensionality decreases and variables are more likely to fit in the hardware cache, thus leveraging the impact of an extensive memory access. A tiling technique is exploited in order to generate a number of independent tasks that can be concurrently solved. Due to iterative nature of inner BCD solver, different running times are expected for the solution of subproblems: to overcome this disadvantage we adopted manager/workers pattern that ensures run-time distribution of independent tasks among POSIX threads. A number of computational threads (workers) is initialized and put on wait on a shared task queue, while a monitor thread (master) is responsible to extract initial data and to collect computed solutions for each subproblem. Mutex-protected queues collect both task input and output results, and therefore, two different queues are present in the implementation:

- a job queue: a single manager is the producer of the queue elements, while all workers are consumers;
- a results queue: in this case each worker fills the queue with results of assigned subproblems, while the manager is responsible to insert them in the overall segmentation variables (u, s, z).

Both cases can be handled by the same implementation that provides:

- a thread-safe interface for insert/remove operations;
- a signaling mechanism for the communication of available resources.

We provide a simple C++ class that stores resources in a private $std :: queue < T >$ variable, while exposing only two methods **push** and **pop** for resource insertion and removal, respectively. This implementation can be used in conjunction of POSIX Threads, since additional private members are present:

- a mutex variable of type *pthread_mutex_t*, used as safeguard for the shared resource;
- a condition variable of type *pthead_cond_t*, associated to previously mentioned mutex, for signaling procedures.

Such implementation choice allows for a mutually exclusive access to internal queue in multi-threading environment. Moreover, through the adoption of a condition variable, producer threads can communicate information about the state of shared data: for example to signal that a queue is no longer empty. In order to provide a reliable queue implementation even in presence of exceptions, RAII (Resource Acquisition Is Initialization) programming idiom is adopted when locking/unlocking operations are executed on a mutex. Job queue is used to communicate both commands and data from master to workers: in this implementation, only two basic job types are used. First job type contains a complete description of one of the tasks (references to subproblem local data, objective function parameters and algorithm parameters). A second type of job is used by master thread in order to ensure the clean termination of workers threads. Each worker thread is structured as a while loop: as long as the thread can pick a subproblem description, it solves it and puts the results on results queue; when a termination job is picked, the thread exits.

We tried our methods on images of cracks taken in tunnels in Greece and backscattered electron images of concrete samples. In the former case the challenge was to reconstruct the whole structure avoiding the effect of the noise and the environmental conditions (i.e. lights). In the latter the aim was to detect the structure despite the complex texture in the background.

Table 5.1: Run time comparisons for a single image

	time (s)
sequential	13.184868
parallel 24 cores	1.056701
parallel 48 cores	0.5915374

In both cases the structures have been detected correctly (see figure 5.10,5.11). In order to reduce the execution time and to provide an automatic procedure we

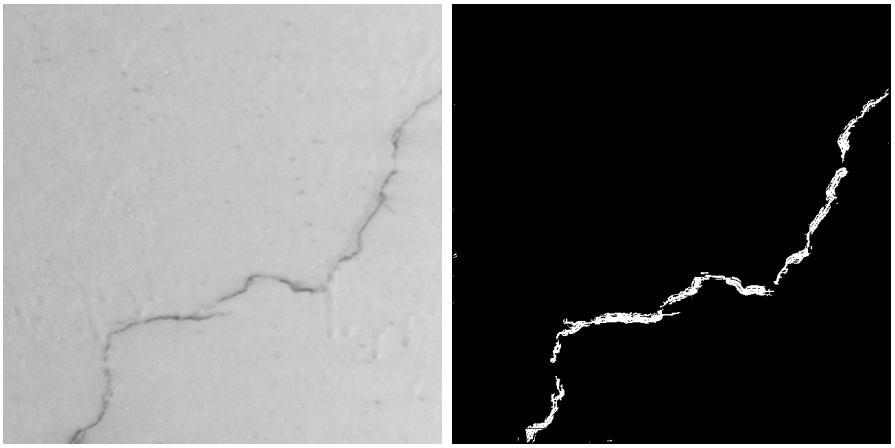


Figure 5.10: Crack on a concrete wall

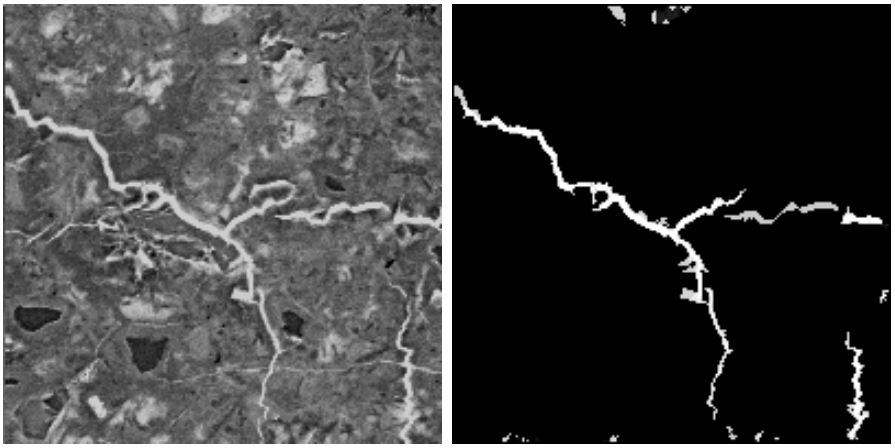


Figure 5.11: Microcracks on a BSE microscopy of concrete materials

tested a sequential implementation against a parallel one based on the OpenMP framework that implements two strategies for collaboratively executing a program on an environment composed by devices of different types (aka heterogeneous architectures). The experiments were performed both on a commodity PC and on a High Performance Computing cluster. A sequential version was executed on a workstation equipped with a processor Intel (R) Xeon CPU E6-79 at 3.40 GHz with 32GB of RAM and total number of cores 12, running an Ubuntu 18.04 operating system. The parallel version based was executed on a heterogeneous cluster equipped with x86-64 processors, running a CentOS 7.6 operating system. Overall, the results show a significant reduction in the execution time with respect to the sequential algorithm.

5.2.2.1 Evaluation Criteria

In order to compare and analyse the results, quantitative measures as long as ground-truth are required. Computing a quantitative criterion requires a ground-truth, i.e. a certified image of the expected result. Our algorithm provides a grey-level image which has a high intensity for curvilinear structures and a low intensity for any other structures. The natural ground-truth for such a feature is a binary image composed of white pixels for curvilinear structures and black pixels, for the rest. Quantifying the intensity of a grey-level intensity feature with a binary ground-truth requires a thresholding procedure. All the images are thresholded at all its gry-level values, resulting in several binary images. Each of these binary images are compared pixel-wise to the ground-truth and the total number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are computed (see Table 5.2)

Table 5.2: Classification of the errors according to the value of the result and the ground-truth

		Value of the binary result	
		0	1
Value of the ground-truth	0	TN	FP
	1	FN	TP

The sum of the TP and FN is the total number of positives (P_{GT}) in the ground-truth whereas the sum of the TN and FP is the total number of negatives (N_{GT}) in the ground-truth. Images containing curvilinear structures are ususally sparse, which means that $P_{GT} << N_{GT}$. Consequently, the number of FP and TN are potentially much higher than the maximum number of TP and FN. To present meaningful results, we define the false positive rate (FPR), true positive rate (TPR), false neegative rate (FPR) and true negative rate (TNR) as follows:

$$\text{FPR} = \frac{FP}{P_{GT}} \quad \text{TPR} = \frac{TP}{P_{GT}}$$
$$\text{FNR} = \frac{FN}{P_{GT}} \quad \text{TNR} = \frac{TN}{P_{GT}}$$

the TPR is also called the **sensitivity** while the FPR is also called the **fall-out**. The closer the TPR to 1 and FPR to 0, the best result. It is important

to note that with these definitions FPR can exceed 1. For example, if $FPR = 2$, the evaluted method detected twice as many false positives as possible true positives. Based on these error measures, several quantitative similarity criteria have been proposed. The widely used criteria to compare segmentation results is the **accuracy**:

$$\text{ACC} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{5.2}$$

The closer this coefficient is to 1, the more similar the result to the ground-truth. Another important metric is **precision** as

$$\text{PPV} = \frac{TP}{TP + FP}. \tag{5.3}$$

Precision expresses how many correct positive predictions the classifier have made; i.e., how many actual cracked areas exist among the classifiers' suggestions as cracked. The harmonic mean of **precision** and **sensitivity** is the $F1$ score,

$$\text{F1} = \frac{2TP}{2TP + FP + FN}. \tag{5.4}$$

The TPR and FPR are representative of one threshold of the evaluated result. To obtain a global vision of a grey-level result, a Receiver Operating Characteristic (ROC) curve is usually computed. A ROC curve is the curve of the TPR against the FPR at every grey-level value of the evaluated result. The closer the curve to the point $[0, 1]$ ($FPR = 0$ and $TPR = 1$) the more similar the result is to the ground-truth.

Table 5.3: Validation results of the proposed computer vision method using different types of metrics

Method	ACC	PPV	TPR	F1
Our method	0.643	0.727	0.730	0.451
CNN	0.637	0.720	0.720	0.494

Our method outperforms the method described in [260] (see Table 5.3), but all these criteria are based on the hypothesis that $P_{GT} \simeq N_{GT}$ which is clearly untrue for sparse images of curvilinear structures. To cope with this problem, a similarity criterion specially designed to deal with unbalanced classes should be

used instead.

5.3 Digital Twin Model and 3D Point Cloud Segmentation

Building Information Modeling (BIM) is an up-to-date modeling concept involving the generation and the management of a three-dimensional (3D) digital representation of physical and functional characteristics of a facility during its entire life-cycle. These digital representations, including geometric and semantic aspects, are called Building Information Models (BIM). Building Information Models are usually used as shared data and knowledge resources to support planning, construction, management activities. In particular, 3D geometric objects, such as walls, columns, and slabs, including their interrelations and life-cycle attributes, are defined within a building information model. One main advantage of the BIM concept is the opportunity to virtually construct the facility prior to its actual physical construction to reduce uncertainty, improve safety, and work out problems as well as to simulate and analyse potential impacts. Another important aspect of BIM is the integration of facility maintenance information, such as preventive maintenance/inspection schedules and intervals, specification and manuals, and as-is performance data, such as current condition and damage states. BIM is often associated with the Industry Foundation Classes (IFCs), which are a data structure for representing complex building information. The IFCs have been developed by **buildingSMART** as a neutral, non-proprietary and open standard for sharing BIM data. Currently, the IFC standard predominantly supports building construction rather than civil infrastructure, such as roads, bridges and tunnel constructions. However, there are initiatives that undertake efforts to extend the IFC and develop an IFC-based model for roads, bridges and shield tunnels. The IFC-Bridge model, for example, integrates information on the general structure of bridges, the complete geometry definition, technological definitions, materials associations (concrete, steel, wool, etc.), prestressing information, and process control. Since BIM could be seen as the civil and building engineering advancement of the Computer-Aided Design (CAD) technology, it is currently predominantly implemented in the design phase of an infrastructure facility. Designers use BIM software tools to create a virtual 3D

model - an *as-designed model* - of a building, bridge, or tunnel. As the main idea of BIM is to use this information model during the entire life-cycle, the as-designed model is supposed to be converted to an *as-built model*, and finally to an *as-damaged model*. While the as-built model contains actual BIM data after the infrastructure facility has been constructed, the as-damaged model also includes damage that has been identified during inspection procedures. However, today very few existing infrastructure facilities have a complete as-designed model available. So the question arises, **how to create an as-built and as-damaged building information model based on the existing facility without having an initial as-designed model ?**

A BIM is 3D model with objects defined as building elements and classified into different categories such as openings, footings and vertical structure, horizontal structure, roof structure. BIM is not only a 3D model as it combines geometric information and a database.

Despite well established BIM processes for new buildings, the majority of existing buildings is not maintained, refurbished or deconstructed with BIM yet.

The creation of an as-built 3D model requires the acquisition of the as-is conditions of the building. The scanner laser is widely used to achieve this goal because of its fast acquisitions and its accuracy. Point clouds also provide a very high level of geometric information. Unfortunately, the creation of as-built 3D models from point clouds remains currently largely a manual process because of the huge amount of data and because of processes, which are difficult to automate.

The creation of as-built BIM involves 3 tasks: geometric modelling of building elements, object category and material properties assignment as well as object relationship modelling. We focus on geometric modelling based on a segmentation approach.

Fully three-dimensional scanners are now widely available. With scanners such as Light Detection and Ranging (LIDAR) and Microsoft Kinect, 3D point clouds can be easily acquired for different purposes. A point cloud is a collection of points that provide a discrete non-parametric representation of a 3D surface. The points have three Cartesian coordinates that define their positions with respect to some fixed coordinate frame.



Figure 5.12: Mckanes Falls bridge

Data Acquisition and Preprocessing

There exist several ways to obtain point cloud data. Firstly, one may use various range imaging sensors that are capable of measuring distances. Secondly, if a parametric or a dense model of a surface is available a point cloud could be sampled from this model.

Range Imaging Sensors Range imaging sensors operate according to several different physical principles: **structured light** and **time-of-flight** are the most widely used in the robotics context. These sensors deliver 2D images that encode distances from scene points to a fixed point (usually the optical center of the acquisition device). Using known calibration parameters, it is possible to project these points into 3D space, thereby obtaining a point cloud. An important property of point cloud data acquired this way is that they represent a surface as observed from a single viewpoint. This implies that the objects and structures in the foreground occlude the background, making the surfaces there partially unobservable. Despite the fact that the points have 3D coordinates, they do not capture the complete 360 view of a scene and are often times called $2\frac{1}{2}$D.

Laser scanning, also known as light detection and ranging (LiDAR), has been used to quickly and accurately acquire three-dimensional (3D) topographic data of visible surfaces. The fundamental principle of LiDAR involves a laser beam to measure the distance from the instrument to a surface of an object based on the time of travel between signal transmission and reception called a laser pulse. The output datasets consisting of $x-, y-, z-$coordinates associated with other

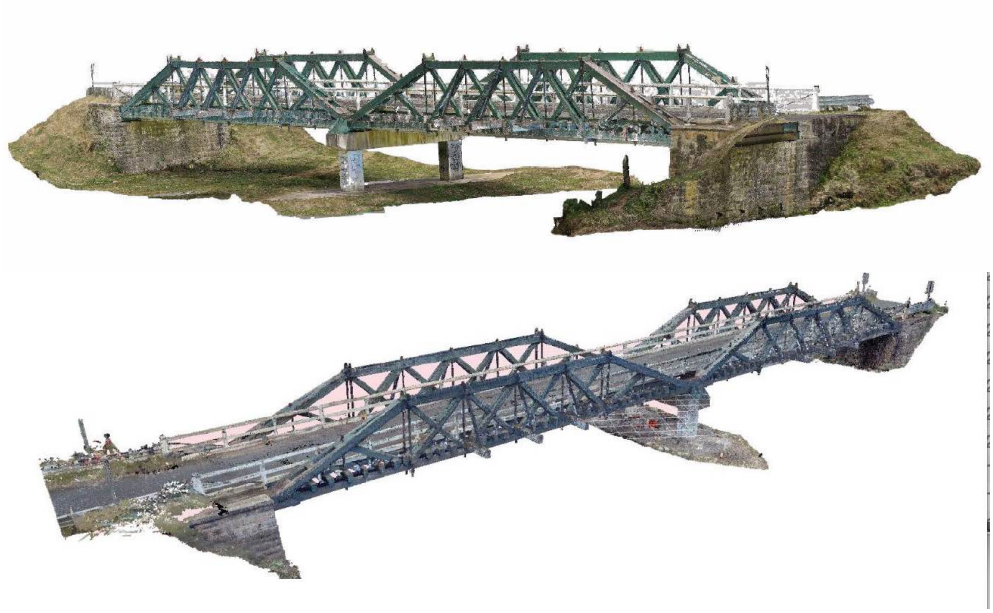


Figure 5.13: UAV photogrammetry vs Terrestrial Laser Scanning

attributes are commonly referred to as a 3D point cloud. One of the LiDAR units is a terrestrial laser scanner (TLS), the laser sensor operating from the ground capturing great details of surfaces of objects with millimetre accuracy. Terrestrial laser scanning has the ability to capture visible surfaces accurately, quickly and efficiently. Laser scanning has been widely used in many civil engineering applications and recently in construction projects for creating as-built building information model.

Over the past few years, advances in the field of 3D imaging have led to manufacturing inexpensive sensors and mainstreaming their use in consumer products (e.g. Kinect, Structure Sensor, RealSense, etc). Microsoft Kinect is an RGB-D sensor. It consists of an RGB colour camera and an infrared (IR) depth sensor and simultaneously provides an RGB and a depth image for a scene. This sensor generates three outputs, namely, IR, RGB, and depth images. The IR projector projects a known IR pattern onto the scene and based on the pattern's deformation as captured by the IR camera, the depth is determined. It is essentially designed for indoor use because the IR component of the sunlight alters the known IR pattern when the object under inspection is exposed to direct sunlight and causes all the captured depth values to be valued wrongly. However, if the object is in shadow (i.e., indirect sunlight), it can obtain the depth infor-

mation accurately. Furthermore, Kinect can provide depth images in complete darkness and, tested under different light conditions, can provide depth images in the presence of artificial light sources such as typical streetlights. [169] describes an autonomous approach for detecting and quantifying defects in pavements using Microsoft Kinect to collect data. The system applies Otsu's method to the histogram of the normalized depth values in order to discriminate between the depressed and flat regions. Using this approach various pavement defects, including patching, severe cracks, and potholes were detected robustly without any need for training and were quantified accurately. Several field experiments were carried out under different light conditions to evaluate its capabilities and its imitations. The tests clearly demonstrate its superior features compared with the traditional pavement-evaluation approaches. Furthermore, it is fully functional at night without any need for a light source, which makes it ideal for night- time condition assessment of structures.

Model Sampling Recent advances in point cloud registration and surface reconstruction make it possible to perform 3D mapping of the environment in real time [233, 331]. Tools like KinectFusion track the motion of an RGB-D camera with respect to the model built so far, simultaneously extending it with the newly acquired data. The model is typically represented with the volumetric truncated signed distance function (TSDF). Using a marching cubes type algorithm it is relatively easy to render the model into a mesh or a point cloud. The resulting point cloud is truly three-dimensional because it incorporates observations from multiple viewpoints. Depending on the number and arrangement of the incorporated views, it is still possible that some surfaces are occluded. However, it is reasonable to assume that in the region of interest, the objects and structures were observed from multiple viewpoints and thus, their models are complete. This makes the conventional 2D image-based processing methods inapplicable.

Many different kinds of preprocessing can be applied to point cloud data. The most known are **downsampling** and **computation of basic geometric features of surface**.

Downsampling The number of points in a cloud may be huge. A scene captured by a Kinect camera at standard resolution contains 300000 points. This amount of data may be hard to process, especially if real-time performance is

required. Therefore, it is often desirable to downsample point cloud data in order to reduce Its amount and retain the quality as much as possible.

Voxelization is a simple method that may be used to downsample a point cloud. A grid is superimposed on 3D Euclidean space; each of its cells is a unit cubic volume called *voxel*. The points of the original cloud that belong to the same voxel are approximated by their centroid. By varying the resolution of the grid, different degrees of downsampling may be achieved. Besides from downsampling the data, voxelization serves two additional purposes. Firstly, a certain degree of smoothing is achieved by suppressing high-frequency noise in point coordinates. Secondly, the grid induces relations between voxels. Indeed, two voxels may share a face, an edge or a corner; they are assumed to be adjacent in such case. This make neighbourhood computation a cheap and straightforward procedure.

Normal and Curvature Estimation A point, when viewed in isolation, is described by its Euclidean coordinates. Two additional attributes - *normal orientation* and *curvature*- may be attached when it is considered as part of the surface from which it was sampled. The former describes the plane tangential to the surface at the point; the latter captures the amount of variation in the surface around the point. These attributes are of an importance for various point cloud visualization and processing tasks; shading and rendering, denoising, feature detection and segmentation are just a few of them.

There exist a number of approaches to normal estimation [181]. The classical method consists of approximating the tangential plane by means of the first order 3D plane fitting and is based on the theory of Principal Component Analysis (PCA). Given a point p_0 and a set of its neighbours $\mathcal{P} = \{p_1, \ldots, p_k\}$, the covariance matrix is defined as

$$\mathcal{C} = \frac{1}{k+1} \sum_{i=0}^{k} (p_i - \overline{p})(p_i - \overline{p})^T, \tag{5.5}$$

where $\overline{p}$ is the centroid of $\mathcal{P} \cup \{p_0\}$. The eigenvectors of this matrix form an orthogonal coordinate frame and correspond to the principal components of $\mathcal{P}$. Thus, the eigenvector associated with the minimal eigenvalue defines the normal of the plane fitted to the neighbourhood.

A few different ways to compute the curvature are available. [274] proposed a simple method which makes use of the eigendecomposition involved in normal

computation. He notes that the eigenvalues λ_j of the covariance matrix $\mathcal{C}$ approximate surface variations around the point p_0. Therefore, the ratio between the minimum eigenvalue λ_0 and the sum of all eigenvalues approximates the curvature. Formally, the curvature σ is defined by

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{5.6}$$

5.3.1 Datasets

Recently, more point cloud datasets have been introduced. they can be classified into two categories: **Indoor** datasets which are captured by Kinect and **outdoor** datasets which are captured by laser scanners such as LIDAR. The use of public datasets allows us to compare different approaches and gives insight into the advantages and disadvantages of these methods.

Cornell RGBD dataset [13]: This dataset has 52 labeled indoor scenes of point clouds with RGB values (24 labeled office scenes and 28 labeled home scenes). Point cloud data are created from original RGB-D images using RGBD-SLAM [107]. The dataset composed from about 550 views, having 2945 segments labeled with 27 object classes.

VMR-Oakland dataset [336]: This dataset contains labeled point cloud data collected from a moving platform around CMU campus. The points were collected using laser scanner and are saved in text format, three real valued coordinates of each point are written in each line. The training, validation and testing data are also available.

KITTI dataset [133]: This dataset includes a large number of unorganized point clouds that were captured by a **Velodyne** laser scanner. It consists of manually annotated ground truth bounding boxes for outdoor objects such as cars, pedestrians, trams, trucks and cyclists.

Robotic 3D Scan Repository [1]: This repository provides collection of 3D point cloud datasets for both indoor and outdoor environments. This is the huge collection of 3D point cloud data and can be used not only for segmentation, but also for different purposes. However, these datasets have not been labeled and they also may need a preprocessing step before using them as input for segmenting algorithms.

5.3.2 Methods

The **segmentation** is a fundamental step in processing 3D point clouds. Given a set of point, the objective of the segmentation process is to cluster points with similar characteristics into homogeneous regions. This process could be helpful for analyzing the scene in various aspects such as locating and recognizing objects, classification and feature extraction. The general way to decompose a 3D model into functionally meaningful regions is to build a graph from the input mesh and cluster it by using information such as normal direction, smoothness or concavity along boundaries. Several methods have been proposed for this problem: convex decomposition, watershed analysis, hierarchical clustering, region growing and spectral clustering. Many of these approaches have been widely used to segment point cloud data, especially in region-based methods [291, 276, 262]. Segmenting objects in 3D point clouds is not a trivial task. The point cloud data are usually noisy, sparse and unorganized. The sampling density of points is also typically uneven due to varying linear and angular rates of the scanner. In addition the surface shape can be arbitrary with sharp features and there is no statistical distribution pattern in the data. Moreover, due to the limitations of the 3D sensors, the foreground is often highly entangled with the background. These problems present a difficult challenge when designing a segmentation algorithm.

In this section we discuss the methodologies have been suggested for the segmentation of 3D point clouds. We categorized them into five classes: **edge-based methods**, **region-based methods**, **attributes-based methods**, **model-based methods** and **graph-based methods**. Fundamentally, there are two basic approaches: a **model-driven** and a **data-driven** approach. The first one uses purely mathematical model and geometric reasoning techniques such as region growing or model fitting in combination with robust estimators to fit linear and non-linear models to point cloud data. This approach allows fast running time and achieves good results in simple scenarios. However, it has some limitations due to the difficulty to choose the size of the fitting model and the extremely high sensitivity to noise. The second one extracts 3D features from data and uses machine learning techniques to learn different classes of object types and classify acquired data. In complex scenes machine learning techniques outperform techniques based on geometric modelling. The reason is due to noise, uneven density and occlusions in point cloud. However, they are usually slow and rely on the

result of feature extraction process.

Edge-based methods . Edges describe the characteristics about the shape of objects. Edge-based methods detect the boundaries of several regions in the point clouds to obtain segmented regions. The principle of these methods is to locate the points which have rapid change in the intensity (see subsection 3.3.3). Bhanu et al. [31] proposed an edge detection technique by computing the gradient, fitting 3D lines to a set of points and detecting changes in the direction of unit normal vectors on the surface. Jiang [117] presented a fast segmentation method using scan line grouping techniques. Scan lines of the range image are split into curves and they are then clustered to represent surfaces. Compared to [31], this method is advantageous in both quality and running time. But it is only suitable for range images and not good for uneven density point clouds. In [280] authors proposed a new edge detection strategy by extracting close contours from a binary edge map for fast segmentation. Although edge-based methods allow fast segmentation, they have accuracy problems because all of them are very sensitive with noise and uneven density of point clouds, a situation that commonly occur in point cloud data.

Region-based methods . Region-based methods use neighbourhood information to combine nearby points that have similar properties to obtain isolated regions. They are more accurate to noise than edge-based methods, but have problem with over and under segmentation and determining region borders accurately. We divide region-based methods into two categories: **seeded-region** (or bottom-up) methods and **unseeded-region** (or top-down) methods.

Seeded-region methods start the segmentation process by choosing a number of seed points, then from these points, each region will grow by adding neighbour points if they satisfy certain criterion or compatibility thresholds. The initial algorithm was introduced by Besl [30] and includes two steps: identification of the seed points based on the curvature of each point and growing them based on predefined criteria such as proximity of points and planarity of surfaces. A drawback of this method is it is very sensitive to noise and is time consuming. Several subsequent works proposed improvements to this initial method. Koster [188] presented an algorithm that generates an irregular graph pyramid to store rela-

tive information between regions. This information is used to compare and merge adjacent regions. The work of Rusu et al. [276] used seeded-region methods based on smoothness constraint as described in [263]. In [193] Tovari introduced a region growing method for airborne laser data. This approach proposed a method for growing the seed points based on their normal vector and its distance to the growing plane. Pu [262] adopted the planar surface growing algorithm [321] for segmenting terrestrial laser data. Many important properties of point cloud data were retrieved from the segments to recognize potential building features. Ning [235] proposed a method that includes two stages as rough and detail segmentation. Rough segmentation is used to extract main objects in the scene based on the consensus of normal vector in the same plane. Detailed segmentation is used as a refined process to extract finer information object components. The work of Dorninger [100], reduced the time complexity by using a sequential implementation of the clustering algorithm. This method segments the original points by hierarchical clustering and coarse contour information. Seeded-region approaches are highly dependent on selected seed points. Inaccurate choosing seed points will affect the segmentation process and can cause under and over segmentation. Choosing seed points as well as controlling the growing process is time consuming. The segmented results could be sensitive to the chosen compatibility thresholds. Another difficulty is to decide whether to add points in a given region, since the decision is done locally which is susceptible to noise.

unseeded-region methods , on the contrary, base on the top-down approach. First, all points are grouped into one region. Then the subdivision process starts to divide it into smaller regions. Chen [76] used this method in guiding the process of clustering planar regions to reconstruct complete geometry of architectural buildings. A limitation of this method is it may have over segmentation and it does not perform well when segment other objects such as trees. The main difficulty of unseeded-region methods are to decide where and how to subdivide. Another limitation of theses methods is that they require a large amount of a prior knowledge (e.g. object models, number of regions, etc.) which are usually unknown in complex scenes.

Attributes-based methods . These methods consist in point clouds clustering based on attributes that are previously computed. The main limitation of

these approaches is they are highly dependent on the quality of derived attributes. Moreover, they should be computed precisely to produce the the best separation among different classes. Briosca [32] introduced a new strategy for segmentation of terrestrial laser point clouds by using unsupervised clustering approach and fuzzy algorithms. This method adapts parameters of fuzzy algorithms to use in combination with cluster merging method. The method is promising, but it relies on choosing correct parameters and is time consuming. Filin [120] proposed a methodology for clustering laser data surfaces. It can cope with the varying point density and operates on the laser points directly without rasterization. An improvement of this approach can be found in [120].This work proposed a segmentation method based cluster analysis in a feature space. In this method the normal vectors are derived using a neighbourhood system called slope adaptive. Neighbourhood among the measured points is defined using attributes of point cloud data e.g. distance, point density and horizontal or vertical point distribution. Then, the slopes of the normal vector in each directions and the height difference between the point and its neighbourhood are used as the clustering attributes. This method can eliminate the influence of outliers or noise. Vosselman [320] used $3D$ version of the well known Hough transform for segmentation of planar surfaces in laser point cloud data. In this method each point is redefined as a plane in the $3D$ attribute space. Authors showed that this method successfully extracts planar faces from the irregularity distributed point clouds, but it sometimes leads to over segmentation results. Attributes-based methods are the robust approach for grouping points into homogeneous regions. Their results are flexible and accurate. However, they rely on the definition of neighbourhood between points and the point density of point cloud data. Another limitation is that they are time consuming especially when dealing with multidimensional attributes.

Model based methods . Model based methods use geometric primitive shapes (e.g. sphere, cone, plane and cylinder) for grouping points. The points which have the same mathematical representation are grouped as one segment. Fisher [121] introduced a well-known algorithm called RANSAC (RANdom Sample Consensus). RANSAC is a robust model and is used to detect mathematical features like straight lines, circles etc. This method is now the state of the art for model fitting. In $3D$ point cloud segmentation, many subsequent works have inherited

this initial algorithm. Schnabel et al. [284] proposed an algorithm that used RANSAC for segmenting both mesh and point cloud data. It works well with outliers and with high noisy data, but it does not scale well on the size of the input point clouds and of the included shapes. To expand the restriction of primitive shapes, Gelfand et al. [134] presented in a method to detect slippable shapes. Slippable shapes are defined as rotationally and translationally symmetrical shapes and include: sphere, helix, plane, cylinder, linear extrusions and surfaces of revolution. This idea can be used to segment point cloud data containing complex shape structure by merging initial slippable surfaces. However, its accuracy relies on the selection of the size of the initial patches, which is hard to determine. Tarsha-Kurdi [306] compared RANSAC and $3D$ Hough transform for automatically detect roof planes from point cloud laser data. Despite the limitation encountered in both methods, RANSAC is more efficient in both segmented results and running time. It can process a large amount of input data in negligible time. In the other hand, $3D$ Hough transform is slower and more sensitive to the segmentation parameters values. The work by Li et al.l [197]. presented an algorithm for globally consolidating the results obtained by RANSAC method. In this approach RANSAC is used for local fitting of primitives. The global coupling corrects the primitives obtained in the local RANSAC stage and brings them to a more precise global alignment. This technique could be used to refine the parameters of the fitted primitives when segmenting point clouds. Model based methods have purely mathematical principles. They are fast and robust with outliers. The main limitation of these methods is their inaccuracy when dealing with different point cloud sources.

Graph-based methods . Graph-based methods consider the point clouds in terms of a graph. In a simple model each vertex corresponds to a point in the data and each edge connects certain pairs of neighbouring points. Graph-based methods are accurate and gain popularity for robotic applications due to its efficiency. A well-known of this approach is Felzenszwalb Huttenlocher algorithm [117]. It operates like Kruskal's algorithm for finding a minimum spanning tree in a graph. Golovinskiy [144] used k-nearest neighbours (KNN) to build a 3D graph on the point cloud. This method introduces a penalty function to encourage smooth segmentation where the foreground is weakly connected to the background and minimize it with min-cut. This method can be run fully automatically or in-

teractively with a user interface, but it requires prior knowledge on the location of the objects to be segmented. In [300], Strom et al. extended graph-based method to segment coloured 3D laser point clouds. By using co-registered sensors, this work proposed a segment union criterion based on colour and surface normals. It can successfully segment coloured point clouds of both indoor and outdoor scenes. The experiment showed that it can run in real time and is considerably more robust than segmenting either laser data alone or colour image alone. The limitation of this method is that it requires a complex sensors system and the segmentation results are sensitive with colour information. Many works on graph-based methods are cast into probabilistic inference model such as Conditional Random Fields (CRF) [192]. Rusu et al. [275] proposed an approach for labelling points with different geometric surface primitives using CRF. Like Nurunnabi [237], this method based on surface segmentation extracts feature descriptor called Fast Point Feature Histograms (FPFH) [275] to encode the local surface geometry around a point. By defining classes of $3D$ geometric surfaces and making use of contextual information using CRF, this method successfully segments and labels 3D points based on their surfaces even with noisy data. Schoenberg et al. [285] presented an algorithm to segment 3D points in dense range data generated from fusion of single optical camera and laser scanner. This method uses Markov Random Field [95] to estimate a 3D point corresponding to each image pixel. Textured dense point clouds are generated from interpolating spare laser range data constrained by an aligned optical image. The weight on graph is computed as a combination of Euclidean distances, pixel intensity differences and angles between surface normals estimated at each point. This method successfully segments point clouds in complex urban environment with near real time performance. Compared with other methods, graph-based techniques can segment complex scenes in point cloud data with noise and uneven density with better results. However, these methods cannot run in real time. Some of them may need offline training step or require special co-registered sensors and camera system.

5.4 Conclusions

Large concrete structures need to be inspected in order to predict future conditions and assess long-term needs, to support investment planning and decision making, and to allocate limited maintenance and rehabilitation [278]. Current procedures for condition and safety assessment of large concrete structures are performed manually leading to subjective and unreliable results, costly and time-consuming data collection, and safety issues. To address these limitations, automated machine vision-based inspection procedures are increasingly proposed by the research community. In this chapter we summarized current achievements and open challenges in vision-based inspection of large concrete structures and we proposed an automatic procedure to detect cracks in images that is robust to changes in light conditions and noise corruption. This technique is based on variational calculus and was implemented on modern HPC architectures in order to process efficiently huge amounts of data collected by robotic machines designed to inspect structures (see subsection 5.2.2 and [247], [248], [249], [246],[250], [251]). Furthermore, we described the state of art of point cloud denoising and segmentation. Point clouds provide fundamental tools for digital representation of 3D surfaces, however these data are corrupted by scanner noise and are not accessible to supervised learning that requires pairs of noisy and clean data. Consequently, it is desirable to be able to denoise the acquired noisy 3D point clouds by solely using the noisy data itself. As future work we would like to apply the techniques described in 3.3.3 to denoise point cloud.

Chapter 6

Future Work

Autonomous mobile robots, such as drones, rovers and legged robots, promise to play a main role in autonomously monitoring of civil structures. These applications require robots to operate for extended periods while performing complex tasks, often in unknown, changing and complicated environments. This brings great challenges, among which is the difficulty of executing a rich repertoire of autonomous, robust, and adaptive behaviors with onboard resources.

More than ten years ago, it was reasonable to anticipate that further improvements to microprocessors would soon close this performance gap. At that time, processor development still kept pace with Moore's law, which predicted a doubling of the number of transistors in a dense integrated circuit about every two years. However, with the end of Moore's law in sight, we can no longer count on this. Hence, we need to explore alternative approaches to both the computing hardware and the data processing algorithms of small, autonomous robots. Processor designers must therefore increasingly look towards alternative means than simply faster clocks to improve computation performance. The processor landscape is becoming much more complex, parallel and specialised, as described well in Sutters online article "Welcome to the Jungle". Pressure to move away from CPUs is even stronger in embedded applications, because here power usage is a critical issue, and parallel, heterogeneous, specialised processors seem to be the only route to achieving the computational performance it needs within power restrictions which will fit real products.

The key to efficient processing which is both fast and consumes little power is to divide computation between a large number of relatively low clock-rate or

otherwise simple cores and to minimise the movement of data between them. A CPU pulls and pushes small pieces of data one by one from and to a separate main memory store as it performs computation, with local caching of regularly used data as the only mechanism for reducing the flow. Programming for a single CPU is straightforward, because any type of algorithm can be broken down into sequential steps with access to a single central memory store, but the piece by piece flow of data to and from central memory has a huge power cost. More efficient processor designs aim to keep processing and the data operated on close together and to limit the transmission of intermediate results. The ideal way to achieve this is a close match between the design of a processor/storage architecture and the algorithm it must run. A GPU certainly has large advantages over a CPU for many computer vision processing tasks. Its SIMT architecture can efficiently run algorithms where the same operation is carried out simultaneously on many different data elements, but in the end a GPU is a processor designed originally for computer graphics rather than vision and AI. Moreover, a joint CPU/GPU architecture is currently needed with substantial data transfer between the two. If we try to look further ahead, we can conceive of processor designs which offer the possibility of a much closer match between architecture and algorithms. With the aim of providing an important new type of processor for AI, Graphcore is a UK company developing 'IPU' processors which comprise thousands of highly interconnected cores on a single chip. As future work, we would like to test our algorithm on this new kind of hardware architecture aiming at building drones specialized in the structural health of civil structures.

In Section 5.3 we described the state of art of point cloud denoising and segmentation. Point clouds provide fundamental tools for digital representation of 3D surfaces, however these data are corrupted by scanner noise and are not accessible to supervised learning that requires pairs of noisy and clean data. Consequently, it is desirable to be able to denoise the acquired noisy 3D point clouds by solely using the noisy data itself. As future work we would like to apply the techniques described in 3.3.3 to denoise this kind of data.

Bibliography

[1] http://kos.informatik.uni-osnabrueck.de/3dscans/.

[2] Ikhlas Abdel-Qader, Sara Pashaie-Rad, Osama Abudayyeh, and Sherif Yehia. Pca-based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*, 37(12):771–778, 2006.

[3] Giovanni Alberti, Guy Bouchitté, and Gianni Dal Maso. The calibration method for the mumford-shah functional and free-discontinuity problems. *Calculus of Variations and Partial Differential Equations*, 16(3):299–333, 2003.

[4] Samuel M Allen and John W Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta metallurgica*, 27(6):1085–1095, 1979.

[5] Luigi Ambrosio. A compactness theorem for a new class of functions of bounded variation. *Bollettino della Unione Matematica Italiana*, 3(4):857–881, 1989.

[6] Luigi Ambrosio. The space sbv () and free discontinuity problems. In *Variational and free boundary problems*, pages 29–45. Springer, 1993.

[7] Luigi Ambrosio and Norman Dancer. *Calculus of variations and partial differential equations: topics on geometrical evolution problems and degree theory*. Springer Science & Business Media, 2012.

[8] Luigi Ambrosio, Loris Faina, and Riccardo March. Variational approximation of a second order free discontinuity problem in computer vision. *SIAM Journal on Mathematical Analysis*, 32(6):1171–1197, 2001.

[9] Luigi Ambrosio, Nicola Fusco, and Diego Pallara. *Functions of bounded variation and free discontinuity problems*. Courier Corporation, 2000.

[10] Luigi Ambrosio and Vincenzo Maria Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[11] Luigi Ambrosio and Vincenzo Maria Tortorelli. On the approximation of free discontinuity problems. 1992.

[12] S Amghibech. Eigenvalues of the discrete p-laplacian for graphs. *Ars Combinatoria*, 67:283–302, 2003.

[13] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, page 0278364912461538, 2012.

[14] Christopher R Anderson. A rayleigh–chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse hermitian matrices. *Journal of Computational Physics*, 229(19):7477–7487, 2010.

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

[16] Roberto Ardon, Laurent D Cohen, and Anthony Yezzi. A new implicit method for surface segmentation by minimal paths: Applications in 3d medical images. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 520–535. Springer, 2005.

[17] Jorge L Ortega Arjona and Graham Roberts. Architectural patterns for parallel programming. 1998.

[18] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

[19] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer Science & Business Media, 2006.

[20] J Gary Augustson and Jack Minker. An analysis of some graph theoretical cluster techniques. *Journal of the ACM (JACM)*, 17(4):571–588, 1970.

[21] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[22] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *Int. J. Comput. Vision*, 82(2):113–132, April 2009.

[23] Guy Barles and Christine Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.

[24] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE transactions on image processing*, 18(11):2419–2434, 2009.

[25] Giovanni Bellettini and Alessandra Coscia. Discrete approximation of a free discontinuity problem. *Numerical Functional Analysis and Optimization*, 15(3-4):201–224, 1994.

[26] Giovanni Bellettini and Alessandro Coscia. Approximation of a functional depending on jumps and corners. 1994.

[27] Serge Belongie, Charless Fowlkes, Fan Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the nyström extension. In *European conference on computer vision*, pages 531–542. Springer, 2002.

[28] Andrea L Bertozzi, Selim Esedoglu, and Alan Gillette. Inpainting of binary images using the cahn–hilliard equation. *IEEE Transactions on image processing*, 16(1):285–291, 2006.

[29] Andrea L Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.

[30] Paul J. Besl and Ramesh C Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):167–192, 1988.

[31] Bir Bhanu, Sungkee Lee, Chih-Cheng Ho, and Tom Henderson. Range data processing: Representation of surfaces by edges. In *Proc. Int. Pattern Recognition Conf*, pages 236–238, 1986.

[32] Josep Miquel Biosca and José Luis Lerma. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):84–98, 2008.

[33] Michael J Black, Guillermo Sapiro, David H Marimont, and David Heeger. Robust anisotropic diffusion. *IEEE Transactions on image processing*, 7(3):421–432, 1998.

[34] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In *European conference on computer vision*, pages 428–441. Springer, 2004.

[35] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987.

[36] Michael Bleyer and Margrit Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *Electronic Imaging 2005*, pages 288–299. International Society for Optics and Photonics, 2005.

[37] Tommaso Boccellari and Franco Tomarelli. Generic uniqueness of minimizer for blake & zisserman functional. *Revista matemática complutense*, 26(2):361–408, 2013.

[38] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

[39] Sébastien Bougleux, Abderrahim Elmoataz, and Mahmoud Melkemi. Discrete regularization on weighted graphs for image and mesh filtering. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 128–139. Springer, 2007.

[40] Blaise Bourdin and Antonin Chambolle. Implementation of an adaptive finite-element approximation of the mumford-shah functional. *Numerische Mathematik*, 85(4):609–646, 2000.

[41] Claire Boyer, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric De Gournay, and Pierre Weiss. On representer theorems and convex regularization. *SIAM Journal on Optimization*, 29(2):1260–1281, 2019.

[42] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[43] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 26–33. IEEE, 2003.

[44] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–1137, 2004.

[45] Yuri Boykov, Vladimir Kolmogorov, Daniel Cremers, and Andrew Delong. An integral solution to surface evolution pdes via geo-cuts. In *European Conference on Computer Vision*, pages 409–422. Springer, 2006.

[46] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[47] Andrea Braides and Gianni Dal Maso. Non-local approximation of the mumford-shah functional. *Calculus of Variations and Partial Differential Equations*, 5(4):293–322, 1997.

[48] Andrea Braides et al. *Approximation of free-discontinuity problems*. Number 1694. Springer Science & Business Media, 1998.

[49] Andrea Braides et al. *Gamma-convergence for Beginners*, volume 22. Clarendon Press, 2002.

[50] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[51] Alexander Brook, Ron Kimmel, and Nir A Sochen. Variational restoration and edge detection for color images. *Journal of Mathematical Imaging and Vision*, 18(3):247–268, 2003.

[52] Michael Buckley and Hugues Talbot. Flexible linear openings and closings. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 109–118. Springer, 2002.

[53] Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88, 2009.

[54] Martin Burger and Stanley J Osher. A survey on level set methods for inverse problems and optimal design. *European Journal of Applied Mathematics*, 16(02):263–301, 2005.

[55] David R Butenhof. *Programming with POSIX threads*. Addison-Wesley Professional, 1997.

[56] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[57] M Carriero, A Leaci, and F Tomarelli. Uniform density estimates for blake & zisserman functional, to appear on commun. *Pure Appl. Anal*, 2010.

[58] Michele Carriero, Antonio Leaci, and Franco Tomarelli. A second order model in image segmentation: Blake & zisserman functional. In *Variational Methods for Discontinuous Structures*, pages 57–72. Springer, 1996.

[59] Michele Carriero, Antonio Leaci, and Franco Tomarelli. Strong minimizers of blake & zisserman functional. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, 25(1-2):257–285, 1997.

[60] Michele Carriero, Antonio Leaci, and Franco Tomarelli. Euler equations for blake and zisserman functional. *Calculus of Variations and Partial Differential Equations*, 32(1):81–110, 2008.

[61] Michele Carriero, Antonio Leaci, and Franco Tomarelli. A candidate local minimizer of blake and zisserman functional. *Journal de mathématiques pures et appliquées*, 96(1):58–87, 2011.

[62] Michele Carriero, Antonio Leaci, and Franco Tomarelli. Image inpainting via variational approximation of a dirichlet problem with free discontinuity:. *Advances in Calculus of Variations*, 7(3):267–295, 2014.

[63] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.

[64] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017.

[65] Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadegh Mahmoudkhani, and Oral Büyüköztürk. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):731–747, 2018.

[66] Myung Jin Chae, Tom Iseley, and Dulcy M Abraham. Computerized sewer pipe condition assessment. In *New Pipeline Technologies, Security, and Safety*, pages 477–493. 2003.

[67] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1):89–97, 2004.

[68] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. In *Theoretical foundations and numerical methods for sparse recovery*, pages 263–340. de Gruyter, 2010.

[69] Antonin Chambolle, Daniel Cremers, and Thomas Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences*, 5(4):1113–1158, 2012.

[70] Antonin Chambolle and Gianni Dal Maso. Discrete approximation of the mumford-shah functional in dimension two. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(4):651–672, 1999.

[71] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[72] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[73] Tony F Chan and Luminita A Vese. A level set algorithm for minimizing the mumford-shah functional in image processing. In *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 161–168. IEEE, 2001.

[74] Bernard Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM (JACM)*, 47(6):1028–1047, 2000.

[75] Jeff Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In *Problems in analysis*, pages 195–200. Princeton University Press, 2015.

[76] Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3):223–236, 2008.

[77] HD Cheng, XJ Shi, and C Glazier. Real-time image thresholding based on sample space reduction and interpolation approach. *Journal of computing in civil engineering*, 17(4):264–272, 2003.

[78] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[79] Laurent Cohen. Minimal paths and fast marching methods for image analysis. In *Handbook of mathematical models in computer vision*, pages 97–111. Springer, 2006.

[80] Laurent D Cohen and Ron Kimmel. Global minimum for active contour models: A minimal path approach. *International journal of computer vision*, 24(1):57–78, 1997.

[81] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[82] Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE transactions on pattern analysis and machine intelligence*, 31(8):1362–1374, 2008.

[83] Daniel Cremers. Dynamical statistical shape priors for level set-based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, 2006.

[84] Daniel Cremers, Stanley J Osher, and Stefano Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, 2006.

[85] Gianni Dal Maso. *An introduction to* Γ*-convergence*, volume 8. Springer Science & Business Media, 2012.

[86] Barrie Dams, Sina Sareh, Ketao Zhang, Paul Shepherd, Mirko Kovac, and Richard J Ball. Aerial additive building manufacturing: three-dimensional printing of polymer structures using drones. *Proceedings of the Institution of Civil Engineers-Construction Materials*, 173(1):3–14, 2020.

[87] Piali Das, Olga Veksler, Vyacheslav Zavadsky, and Yuri Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27(1):206–219, 2009.

[88] E De Giorgi, M Carriero, and A Leaci. Existence theorem for a minimum problem with free discontinuity set. *Ennio De Giorgi*, page 654, 1989.

[89] Ennio De Giorgi. Free discontinuity problems in calculus of variations. *Frontiers in pure and applied Mathematics, a collection of papers dedicated*

to JL Lions on the occasion of his 60th birthday, R. Dautray ed., North Holland, 1991.

[90] Ennio De Giorgi and Tullio Franzoni. Su un tipo di convergenza variazionale. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti*, 58(6):842–850, 1975.

[91] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

[92] Andrew Delong and Yuri Boykov. A scalable graph-cut algorithm for nd grids. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[93] Xavier Desquesnes, Abderrahim Elmoataz, and Olivier Lezoray. Pdes level sets on weighted graphs. In *2011 18th IEEE International Conference on Image Processing*, pages 3377–3380. IEEE, 2011.

[94] E DGA. De giorgi, l. ambrosio, un nuovo funzionale del calcolo delle variazioni. *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur*, 82:199–210, 1988.

[95] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *NIPS*, volume 5, pages 291–298, 2005.

[96] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[97] EW Dijkstra. Some theorems on spanning subtrees of a graph. 1960.

[98] Chris HQ Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 107–114. IEEE, 2001.

[99] Julia A Dobrosotskaya and Andrea L Bertozzi. A wavelet-laplace variational technique for image deconvolution and inpainting. *IEEE Transactions on Image Processing*, 17(5):657–663, 2008.

[100] P Dorninger and C Nothegger. 3d segmentation of unstructured point clouds for building modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(3/W49A):191–196, 2007.

[101] Leo Dorst and Arnold WM Smeulders. Best linear unbiased estimators for properties of digitized straight lines. *IEEE transactions on pattern analysis and machine intelligence*, (2):276–282, 1986.

[102] Peter G Doyle and J Laurie Snell. *Random walks and electric networks*. Mathematical Association of America,, 1984.

[103] Abraham Duarte, Ángel Sánchez, Felipe Fernández, and Antonio S Montemayor. Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic. *Pattern Recognition Letters*, 27(11):1239–1251, 2006.

[104] Pasqua DAmbra and Gaetano Tartaglione. Solution of ambrosio–tortorelli model for image segmentation by generalized relaxation method. *Communications in Nonlinear Science and Numerical Simulation*, 20(3):819–831, 2015.

[105] Noha Youssry El-Zehiry and Leo Grady. Contrast driven elastica for image segmentation. *IEEE Transactions on Image Processing*, 25(6):2508–2518, 2016.

[106] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE transactions on Image Processing*, 17(7):1047–1060, 2008.

[107] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm, and Wolfram Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, volume 180, 2011.

[108] Selim Esedog, Yen-Hsi Richard Tsai, et al. Threshold dynamics for the piecewise constant mumford–shah functional. *Journal of Computational Physics*, 211(1):367–384, 2006.

[109] Selim Esedoglu and Riccardo March. Segmentation with depth but without detecting junctions. *Journal of Mathematical Imaging and Vision*, 18(1):7–15, 2003.

[110] Selim Esedoglu, Steven J Ruuth, and Richard Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 10(3):263–282, 2008.

[111] Lawrence C Evans. Convergence of an algorithm for mean curvature motion. *Indiana University mathematics journal*, pages 533–557, 1993.

[112] Alexandre X Falcao and Jayaram K Udupa. Segmentation of 3d objects using live wire. In *Medical Imaging 1997*, pages 228–235. International Society for Optics and Photonics, 1997.

[113] Alexandre X Falcão, Jayaram K Udupa, and Flavio Keidi Miyazawa. An ultra-fast user-steered image segmentation paradigm: live wire on the fly. *IEEE transactions on medical imaging*, 19(1):55–62, 2000.

[114] Alexandre X Falcao, Jayaram K Udupa, Supun Samarasekera, and Bruce E Hirsch. User-steered image boundary segmentation. In *Medical Imaging 1996*, pages 278–288. International Society for Optics and Photonics, 1996.

[115] Alexandre X Falcão, Jayaram K Udupa, Supun Samarasekera, Shoba Sharma, Bruce Elliot Hirsch, and Roberto de A Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *Graphical models and image processing*, 60(4):233–260, 1998.

[116] Paolo Favaro and Stefano Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–631. IEEE, 2004.

[117] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*.

[118] Xiaobing Feng and Andreas Prohl. Numerical analysis of the allen-cahn equation and approximation for mean curvature flows. *Numerische Mathematik*, 94(1):33–65, 2003.

[119] Giorgio Figliolini, Pierluigi Rea, and Marco Conte. Mechanical design of a novel biped climbing and walking robot. In *ROMANSY 18 Robot Design, Dynamics and Control*, pages 199–206. Springer, 2010.

[120] Sagi Filin. Surface clustering from airborne laser scanning data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):119–124, 2002.

[121] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[122] Lester Randolph Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404, 1956.

[123] Lester Randolph Ford Jr and Delbert Ray Fulkerson. *Flows in networks*. Princeton university press, 2015.

[124] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–225, 2004.

[125] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. In *International conference on medical image computing and computer-assisted intervention*, pages 130–137. Springer, 1998.

[126] Pierre Frankel, Guillaume Garrigos, and Juan Peypouquet. Splitting methods with variable metric for kurdyka–łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, 2015.

[127] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 755–762. IEEE, 2005.

[128] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[129] Stuart Ganan and D McClure. Bayesian image analysis: an application to single photon emission tomography. *Amer. Statist. Assoc*, pages 12–18, 1985.

[130] Cristina Garcia-Cardona, Arjuna Flenner, and Allon G Percus. Multiclass diffuse interface models for semi-supervised learning on graphs. *arXiv preprint arXiv:1212.0945*, 2012.

[131] Cristina Garcia-Cardona, Ekaterina Merkurjev, Andrea L Bertozzi, Arjuna Flenner, and Allon G Percus. Multiclass data segmentation using diffuse interface methods on graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1600–1613, 2014.

[132] Ronald F Gariepy. Functions of bounded variation and free discontinuity problems (oxford mathematical monographs) by luigi ambrosio, nicolo fucso and diego pallara: 434 pp.,£ 55.00, isbn 0-19-850254-1 (clarendon press, oxford, 2000). *Bulletin of the London Mathematical Society*, 33(4):492–512, 2001.

[133] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[134] Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 214–223. ACM, 2004.

[135] Donald Geman and Basilis Gidas. *Image analysis and computer vision*. University of Massachusetts, 1991.

[136] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[137] Stuart Geman and Christine Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2, 1986.

[138] Mariano Giaquinta, Giuseppe Modica, and Jiri Soucek. Cartesian currents in the calculus of variations, i, ii. ergebnisse math. grenzgebiete (iii ser), 37, 38, 1998.

[139] Guy Gilboa and Stanley Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[140] Massimo Gobbino. Finite difference approximation of the mumford-shah functional. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 51(2):197–228, 1998.

[141] Massimo Gobbino and Maria Giovanna Mora. Finite-difference approximation of free-discontinuity problems. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 131(3):567–595, 2001.

[142] Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

[143] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[144] Aleksey Golovinskiy and Thomas Funkhouser. Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*, September 2009.

[145] Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[146] Rafael C Gonzalez and Richard E Woods. Digital image processing. *Nueva Jersey*, 2008.

[147] Leo Grady. Multilabel random walker image segmentation using prior models. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 763–770. IEEE, 2005.

[148] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[149] Leo Grady. Minimal surfaces extend shortest path segmentation methods to 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):321–334, 2010.

[150] Leo Grady and Christopher V Alvino. The piecewise smooth mumford–shah functional on an arbitrary graph. *IEEE Transactions on Image Processing*, 18(11):2547–2561, 2009.

[151] Leo Grady, V Sun, and James Williams. Interactive graph-based segmentation methods in cardiovascular imaging. In *Handbook of mathematical models in computer vision*, pages 453–469. Springer, 2006.

[152] Leo J Grady and Jonathan R Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*, volume 3. Springer, 2010.

[153] Leo John Grady and Eric L Schwartz. *Anisotropic interpolation on graphs: The combinatorial Dirichlet problem*. Citeseer, 2003.

[154] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society: Series B (Methodological)*, 51(2):271–279, 1989.

[155] Luigi Grippof and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization methods and software*, 10(4):587–637, 1999.

[156] Jonathan L Gross and Jay Yellen. *Graph theory and its applications*. CRC press, 2005.

[157] Lin Guo, Kevin Rogers, and Robin Kirkham. A climbing robot with continuous motion. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2495–2500. IEEE, 1994.

[158] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085, 1992.

[159] Norman Hallermann and Guido Morgenthal. Visual inspection strategies for large bridges using unmanned aerial vehicles (uav). In *Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management*, pages 661–667, 2014.

[160] Ghassan Hamarneh, Johnson Yang, Chris McIntosh, and Morgan Langille. 3d live-wire-based semi-automatic segmentation of medical images. In *Medical imaging*, pages 1597–1603. International Society for Optics and Photonics, 2005.

[161] Henk Heijmans, Michael Buckley, and Hugues Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision*, 22(2):107–119, 2005.

[162] Anil Nirmal Hirani. *Discrete exterior calculus*. California Institute of Technology, 2003.

[163] Shigeo Hirose and Hiroshi Tsutsumitake. Disk rover: A wall-climbing robot using permanent. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2074–2079. IEEE, 1992.

[164] Dorit S Hochbaum. The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 325–337. Springer, 1998.

[165] D. Hoiem, A.A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*.

[166] D. Hoiem, A.A. Efros, and M. Hebert. Geometric context from a single image.

[167] Huiyi Hu, Justin Sunu, and Andrea L Bertozzi. Multi-class graph mumford-shah model for plume detection using the mbo scheme. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 209–222. Springer, 2015.

[168] Hiroshi Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1333–1336, 2003.

[169] Mohammad R Jahanshahi, Farrokh Jazizadeh, Sami F Masri, and Burcin Becerik-Gerber. Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor. *Journal of Computing in Civil Engineering*, 27(6):743–754, 2012.

[170] Marie-Pierre Jolly, Hui Xue, Leo Grady, and Jens Guehring. Combining registration and minimum surfaces for the segmentation of the left ventricle in cardiac cine mr images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 910–918. Springer, 2009.

[171] HweeKwon Jung, ChangWon Lee, and Gyuhae Park. Fast and non-invasive surface crack detection of press panels using image processing. *Procedia Engineering*, 188:72–79, 2017.

[172] Yoon Mo Jung, Sung Ha Kang, and Jianhong Shen. Multiphase image segmentation via modica–mortola phase transition. *SIAM Journal on Applied Mathematics*, 67(5):1213–1232, 2007.

[173] S. Kakutani. Markov processes and the dirichlet problem, 1945.

[174] Dongho Kang and Young-Jin Cha. Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geotagging. *Computer-Aided Civil and Infrastructure Engineering*, 33(10):885–902, 2018.

[175] David R Karger. Minimum cuts in near-linear time. *Journal of the ACM (JACM)*, 47(1):46–76, 2000.

[176] David R Karger and Clifford Stein. A new approach to the minimum cut problem. *Journal of the ACM (JACM)*, 43(4):601–640, 1996.

[177] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[178] Bernd Kawohl. From mumford–shah to perona–malik in image processing. *Mathematical methods in the applied sciences*, 27(15):1803–1814, 2004.

[179] Dina Khattab, Hala Mousher Ebied, Ashraf Saad Hussein, and Mohamed Fahmy Tolba. Color image segmentation based on different color space models using automatic grabcut. *The Scientific World Journal*, 2014, 2014.

[180] H Kim, Sung-Han Sim, S Cho, et al. Unmanned aerial vehicle (uav)-powered concrete crack detection based on digital image processing. In *Proceedings of 6th International Conference on Advances in Experimental Structural Engineering, 11th International Workshop on Advanced Smart Materials and Smart Structures Technology*, 2015.

[181] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3206–3211. IEEE, 2009.

[182] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[183] Robert V Kohn and Peter Sternberg. Local minimisers and singular perturbations. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 111(1-2):69–84, 1989.

[184] Vladimir Kolmogorov, Yuri Boykov, and Carsten Rother. Applications of parametric maxflow in computer vision. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[185] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? In *European conference on computer vision*, pages 65–81. Springer, 2002.

[186] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004.

[187] Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE transactions on pattern analysis and machine intelligence*, 29(8):1436–1453, 2007.

[188] Klaus Koster and Michael Spann. Mir: an approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):430–444, 2000.

[189] Karl Krissian, Grégoire Malandain, Nicholas Ayache, Régis Vaillant, and Yves Trousset. Model-based detection of tubular structures in 3d images. *Computer vision and image understanding*, 80(2):130–171, 2000.

[190] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[191] S.H. Kwok and A.G. Constantinides. Fast recursive shortest spanning tree for image segmentation and edge detection. *IEEE Transactions on Image Processing*.

[192] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

[193] Z Lari, A Habib, and E Kwak. An adaptive approach for segmentation of 3d laser point cloud. In *ISPRS Workshop Laser Scanning*, pages 29–31, 2011.

[194] Jan Lellmann and Christoph Schnörr. Continuous multiclass labeling approaches and algorithms. *SIAM Journal on Imaging Sciences*, 4(4):1049–1096, 2011.

[195] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *2009 IEEE 12th International Conference on Computer Vision*, pages 277–284. IEEE, 2009.

[196] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[197] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit: consistently fitting primitives by

discovering global relations. In *ACM Transactions on Graphics (TOG)*, volume 30, page 52. ACM, 2011.

[198] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 303–308. ACM, 2004.

[199] Ronny Salim Lim, Hung Manh La, and Weihua Sheng. A robotic crack inspection and mapping system for bridge deck maintenance. *IEEE Transactions on Automation Science and Engineering*, 11(2):367–378, 2014.

[200] Yi-zhou Lin, Zhen-hua Nie, and Hong-wei Ma. Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 32(12):1025–1046, 2017.

[201] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International journal of computer vision*, 30(2):117–156, 1998.

[202] Jiangyu Liu and Jian Sun. Parallel graph-cuts by adaptive bottom-up merging. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2181–2188. IEEE, 2010.

[203] Jiangyu Liu, Jian Sun, and Heung-Yeung Shum. Paint selection. In *ACM Transactions on Graphics (ToG)*, volume 28, page 69. ACM, 2009.

[204] Yufei Liu, Soojin Cho, Billie F Spencer Jr, and Jiansheng Fan. Automated assessment of cracks on concrete surfaces using adaptive digital image processing. *Smart Structures and Systems*, 14(4):719–741, 2014.

[205] Cristian Lorenz, I-C Carlsen, Thorsten M Buzug, Carola Fassnacht, and Jürgen Weese. Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2d and 3d medical images. In *CVRMed-MRCAS'97*, pages 233–242. Springer, 1997.

[206] Xiyang Luo and Andrea L Bertozzi. Convergence analysis of the graph allen-cahn scheme. Technical report, University of California Los Angeles Los Angeles United States, 2016.

[207] Helmut Lutkepohl. Handbook of matrices. *Computational Statistics and Data Analysis*, 2(25):243, 1997.

[208] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[209] LO Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization*, 33(6):1916–1925, 1995.

[210] Alberto Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1(2):169–182, 1972.

[211] Alberto Martelli. An application of heuristic search methods to edge and contour detection. *Communications of the ACM*, 19(2):73–83, 1976.

[212] Zhaoyi Meng, Ekaterina Merkurjev, Alice Koniges, and Andrea L Bertozzi. Hyperspectral image classification using graph clustering methods. *Image Processing On Line*, 7:218–245, 2017.

[213] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An mbo scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6(4):1903–1930, 2013.

[214] Barry Merriman, James Kenyard Bence, and Stanley Osher. *Diffusion generated motion by mean curvature*, volume 27. Department of Mathematics, University of California, Los Angeles, 1992.

[215] Barry Merriman and Steven J Ruuth. Diffusion generated motion of curves on surfaces. *Journal of Computational Physics*, 225(2):2267–2282, 2007.

[216] Odyssée Merveille, Benoît Naegel, Hugues Talbot, and Nicolas Passat. n d variational restoration of curvilinear structures with prior-based directional regularization. *IEEE Transactions on Image Processing*, 28(8):3848–3859, 2019.

[217] Odyssée Merveille, Hugues Talbot, Laurent Najman, and Nicolas Passat. Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):304–317, 2017.

[218] Fernand Meyer. Minimum spanning forests for morphological segmentation. In *Mathematical morphology and its applications to image processing*, pages 77–84. Springer, 1994.

[219] Elad Mezuman and Yair Weiss. Globally optimizing graph partitioning problems using message passing. In *Artificial Intelligence and Statistics*, pages 770–778. PMLR, 2012.

[220] Luciano Modica. Un esempio di γ-convergenza. *Boll. Un. Mat. Ital. B*, 14:285–299, 1977.

[221] Ugo Montanari. On the optimal detection of curves in noisy pictures. *Communications of the ACM*, 14(5):335–345, 1971.

[222] Federico Monti, Michael M Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. *arXiv preprint arXiv:1704.06803*, 2017.

[223] Vincent Morard, Etienne Decenciere, and Petr Dokladal. Geodesic attributes thinnings and thickenings. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 200–211. Springer, 2011.

[224] Jean-Michel Morel and Sergio Solimini. Variational methods in image segmentation. *Bull. Amer. Math. Soc*, 33:211–216, 1996.

[225] O.J. Morris, M. de J Lee, and A.G. Constantinides. Graph theory for image analysis: an approach based on the shortest spanning tree. *IEE Proceedings F (Communications, Radar and Signal Processing)*.

[226] OJ Morris, M de J Lee, and AG Constantinides. Graph theory for image analysis: an approach based on the shortest spanning tree. In *IEE Proceedings F (Communications, Radar and Signal Processing)*, volume 133, pages 146–152. IET, 1986.

[227] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, 1995.

[228] Eric N Mortensen and William A Barrett. Interactive segmentation with intelligent scissors. *Graphical models and image processing*, 60(5):349–384, 1998.

[229] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 1989.

[230] Laurent Najman and Hugues Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013.

[231] Paolo Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

[232] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[233] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[234] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43(2):445–456, 2010.

[235] Xiaojuan Ning, Xiaopeng Zhang, Yinghui Wang, and Marc Jaeger. Segmentation of architecture shape information from 3d point cloud. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, pages 127–132. ACM, 2009.

[236] Dominikus Noll and Aude Rondepierre. Convergence of linesearch and trust-region methods using the kurdyka–łojasiewicz inequality. In *Computational and analytical mathematics*, pages 593–611. Springer, 2013.

[237] Abdul Nurunnabi, David Belton, and Geoff West. Robust segmentation in laser scanning 3d point cloud data. In *Digital Image Computing Techniques*

and Applications (DICTA), 2012 International Conference on, pages 1–8. IEEE, 2012.

[238] Stanley Osher and Nikos Paragios (eds). *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, 2003.

[239] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[240] Stanley Osher and Ronald P Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational physics*, 169(2):463–502, 2001.

[241] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[242] Stanley Osher and Jianhong Shen. Digitized pde method for data restoration. In *Handbook of Analytic-Computational Methods in Applied Mathematics*, pages 751–771. Chapman and Hall/CRC, 2019.

[243] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[244] Massimiliano Pavan and Marcello Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–145. IEEE, 2003.

[245] Massimiliano Pavan and Marcello Pelillo. Efficient out-of-sample extension of dominant-set clusters. In *Advances in Neural Information Processing Systems*, pages 1057–1064, 2004.

[246] Pellegrino and Stathaki. *Automatic smart concrete quality control using supercomputing*. Resilient Materials 4 Life - Cardiff University, UK United Kingdom.

[247] Pellegrino and Stathaki. *Crack detection on heterogeneous High Performance Computing architectures*. University of Bath, UK United Kingdom, sep 2019.

[248] Pellegrino and Stathaki. *An Automated Procedure For Crack Detection*. Springer International Publishing, 2022.

[249] Pellegrino and Stathaki. *Automatic Crack Detection with Calculus of Variations*. Lecture Notes in Networks and Systems. Springer International Publishing, 2022.

[250] Pellegrino and Stathaki. *A model driven method for crack detection in robotic inspection*. Springer International Publishing, 2022. (in press).

[251] Pellegrino and Stathaki. *A variational method for crack detection in robotic inspection*. Springer International Publishing, 2022. (in press).

[252] Bo Peng, Lei Zhang, David Zhang, and Jian Yang. Image segmentation by iterated region merging with localized graph cuts. *Pattern Recognition*, 44(10):2527–2538, 2011.

[253] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[254] Gabriel Peyré and Laurent D Cohen. Geodesic methods for shape and surface processing. *Advances in Computational Vision and Medical Image Processing*, pages 29–56, 2009.

[255] Brent M Phares, Dennis D Rolander, Benjamin A Graybeal, and Glenn A Washer. Reliability of visual bridge inspection. *Public Roads*, 64(5):22–29, 2001.

[256] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the mumford-shah functional. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1133–1140. IEEE, 2009.

[257] DL Pope, Dennis L Parker, DE Gustafson, and PD Clayton. Dynamic search algorithms in left ventricular border recognition and analysis of coronary arteries. In *IEEE*, 1984.

[258] Robert Clay Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

[259] D Proffitt and D Rosen. Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. *Computer graphics and image processing*, 10(4):318–332, 1979.

[260] E Protopapadakis, C Stentoumis, N Doulamis, A Doulamis, K Loupos, K Makantasis, G Kopsiaftis, and A Amditis. Autonomous robotic inspection in tunnels. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 167–174, 2016.

[261] Eftychios Protopapadakis, Athanasios Voulodimos, Anastasios Doulamis, Nikolaos Doulamis, and Tania Stathaki. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. *Applied intelligence*, 49(7):2793–2806, 2019.

[262] Shi Pu, George Vosselman, et al. Automatic extraction of building features from terrestrial laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):25–27, 2006.

[263] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann. Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):248–253, 2006.

[264] Roberto Rigamonti and Vincent Lepetit. Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 189–197. Springer, 2012.

[265] Bart M Haar Romeny. *Geometry-driven diffusion in computer vision*, volume 1. Springer Science & Business Media, 2013.

[266] Azriel Rosenfeld and Pilar De La Torre. Histogram concavity analysis as an aid in threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):231–235, 1983.

[267] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[268] Sébastien Roy and Ingemar J Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Computer Vision, 1998. Sixth International Conference on*, pages 492–499. IEEE, 1998.

[269] Jacob Rubinstein, Peter Sternberg, and Joseph B Keller. Fast reaction, slow diffusion, and curve shortening. *SIAM Journal on Applied Mathematics*, 49(1):116–133, 1989.

[270] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 31–35. IEEE, 1994.

[271] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[272] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018.

[273] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1605–1614. IEEE, 2006.

[274] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[275] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[276] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[277] Steven J Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *Journal of Computational Physics*, 144(2):603–625, 1998.

[278] Rafael Sacks, Ioannis Brilakis, Ergo Pikas, Haiyan Sally Xie, and Mark Girolami. Construction with digital twin information systems. *Data-Centric Engineering*, 1, 2020.

[279] Philippe Salembier, Albert Oliveras, and Luis Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

[280] Angel Domingo Sappa and Michel Devy. Fast range image segmentation by an edge detection strategy. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 292–299. IEEE, 2001.

[281] Yoshinobu Sato, Shin Nakajima, Hideki Atsumi, Thomas Koller, Guido Gerig, Shigeyuki Yoshida, and Ron Kikinis. 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In *CVRMed-MRCAS'97*, pages 213–222. Springer, 1997.

[282] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, Guido Gerig, and Ron Kikinis. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis*, 2(2):143–168, 1998.

[283] J Savall, Alejo Avello, and Leoncio Briones. Two compact robots for remote inspection of hazardous areas in nuclear power plants. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 3, pages 1993–1998. IEEE, 1999.

[284] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[285] Jonathan R Schoenberg, Aaron Nathan, and Mark Campbell. Segmentation of dense range information in complex urban scenes. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2033–2038. IEEE, 2010.

[286] Kap-Ho Seo and Ju-Jang Lee. Real-time object tracking and segmentation using adaptive color snake model. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, pages 5–pp. IEEE, 2005.

[287] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

[288] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

[289] Jayant Shah. Uses of elliptic approximations in computer vision. In *Variational Methods for Discontinuous Structures*, pages 19–34. Springer, 1996.

[290] Jayant Shah, Homer H Pien, and John M Gauch. Recovery of surfaces with discontinuities by fusing shading and range data within a variational framework. *IEEE Transactions on Image Processing*, 5(8):1243–1251, 1996.

[291] Ariel Shamir. Segmentation and shape extraction of 3d boundary meshes. *State-of-the-Art Report, Proceedings Eurographics*, 2006:137–49, 2006.

[292] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[293] David Shulman and Jean-Yves Herve. Regularization of discontinuous flow fields. In *Proc. workshop on visual motion*, pages 81–86. IEEE Computer Society Press, 1989.

[294] Ali Kemal Sinop and Leo Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[295] Pierre Soille, Edmond J Breen, and Ronald Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 18(5):562–567, 1996.

[296] X Yu Stella, Ralph Gross, and Jianbo Shi. Concurrent object recognition and segmentation by graph partitioning. In *Advances in neural information processing systems*, pages 1383–1390, 2002.

[297] Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM (JACM)*, 44(4):585–591, 1997.

[298] Petter Strandmark and Fredrik Kahl. Parallel and distributed graph cuts by dual decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2085–2092. IEEE, 2010.

[299] Gilbert Strang. Maximal flow through a domain. *Mathematical Programming*, 26(2):123–143, 1983.

[300] Johannes Strom, Andrew Richardson, and Edwin Olson. Graph-based segmentation for colored 3d laser point clouds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2131–2136. IEEE, 2010.

[301] Herb Sutter and Andrei Alexandrescu. *C++ coding standards: 101 rules, guidelines, and best practices*. Pearson Education, 2004.

[302] Arthur Szlam and Xavier Bresson. A total variation-based graph clustering algorithm for cheeger ratio cuts. *UCLA Cam report*, pages 09–68, 2009.

[303] Cheng Tai, Xiaoqun Zhang, and Zuowei Shen. Wavelet frame based multiphase image segmentation. *SIAM Journal on Imaging Sciences*, 6(4):2521–2546, 2013.

[304] Hugues Talbot and Ben Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing*, 25(4):416–425, 2007.

[305] Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)*, 22(2):215–225, 1975.

[306] Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenmeyer, et al. Hough-transform and extended ransac algorithms for automatic detection of 3d

building roof planes from lidar data. In *Proceedings of the ISPRS Workshop on Laser Scanning*, volume 36, pages 407–412, 2007.

[307] Nicolás Garcia Trillos and Dejan Slepčev. Continuum limit of total variation on point clouds. *Archive for rational mechanics and analysis*, 220(1):193–241, 2016.

[308] Andy Tsai, Anthony Yezzi, and Alan S Willsky. Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE transactions on Image Processing*, 10(8):1169–1186, 2001.

[309] James Alexander Tyrrell, Emmanuelle di Tomaso, Daniel Fuja, Ricky Tong, Kevin Kozak, Rakesh K Jain, and Badrinath Roysam. Robust 3-d modeling of vasculature imagery using superellipsoids. *IEEE transactions on medical imaging*, 26(2):223–237, 2007.

[310] Michael Unser, Julien Fageot, and Harshit Gupta. Representer theorems for sparsity-promoting regularization. *IEEE Transactions on Information Theory*, 62(9):5167–5180, 2016.

[311] Michael Unser, Julien Fageot, and John Paul Ward. Splines are universal solutions of linear inverse problems with generalized tv regularization. *SIAM Review*, 59(4):769–793, 2017.

[312] Jasper J van de Gronde, Herman R Schubert, and Jos BTM Roerdink. Fast computation of greyscale path openings. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 621–632. Springer, 2015.

[313] Yves Van Gennip, Andrea L Bertozzi, et al. Gamma convergence of graph ginzburg-landau functionals. *Advances in Differential Equations*, 17(11/12):1115–1180, 2012.

[314] Yves Van Gennip, Nestor Guillen, Braxton Osting, and Andrea L Bertozzi. Mean curvature, threshold dynamics, and phase field theory on finite graphs. *Milan Journal of Mathematics*, 82(1):3–65, 2014.

[315] Olga Veksler. Graph cut based optimization for mrfs with truncated convex priors. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[316] Olga Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision*, pages 454–467. Springer, 2008.

[317] Ben JH Verwer, Lucas J van Vliet, and Piet W Verbeek. Binary and grey-value skeletons: Metrics and algorithms. In *Thinning Methodologies For Pattern Recognition*, pages 323–344. World Scientific, 1994.

[318] Luc Vincent. Minimal path algorithms for the robust detection of linear features in gray images. *Computational Imaging and Vision*, 12:331–338, 1998.

[319] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[320] George Vosselman, Sander Dijkman, et al. 3d building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4):37–44, 2001.

[321] George Vosselman, Ben GH Gorte, George Sithole, and Tahir Rabbani. Recognising structure in laser scanner point clouds. *International archives of photogrammetry, remote sensing and spatial information sciences*, 46(8):33–38, 2004.

[322] Dorothea Wagner and Frank Wagner. Between min cut and graph bisection. In *International Symposium on Mathematical Foundations of Computer Science*, pages 744–750. Springer, 1993.

[323] Song Wang and Jeffrey Mark Siskind. Image segmentation with minimum mean cut. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 517–524. IEEE, 2001.

[324] Jörgen W Weibull. *Evolutionary game theory*. MIT press, 1997.

[325] Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[326] Joachim Weickert. Coherence-enhancing diffusion filtering. *International journal of computer vision*, 31(2):111–127, 1999.

[327] Max Wertheimer. Laws of organization in perceptual forms. 1938.

[328] Joan S Weszka. A survey of threshold selection techniques. *Computer Graphics and Image Processing*, 7(2):259–265, 1978.

[329] Joan S Weszka and Azriel Rosenfeld. Threshold evaluation techniques. *IEEE Transactions on systems, man, and cybernetics*, 8(8):622–629, 1978.

[330] John S Weszka and Azriel Rosenfeld. Histogram modification for threshold selection, 1979.

[331] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012.

[332] Gerhard Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer Science & Business Media, 2012.

[333] Andrew Witkin. Scale-space filtering: A new approach to multi-scale description. In *ICASSP'84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 150–153. IEEE, 1984.

[334] Zhenyu Wu and R Leahy. Tissue classification in mr images using hierarchical segmentation. In *Nuclear Science Symposium, 1990. Conference record: Including Sessions on Nuclear Power Systems and Medical Imaging Conference, 1990 IEEE*, pages 1410–1414. IEEE, 1990.

[335] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 15(11):1101–1113, 1993.

[336] Xuehan Xiong, Daniel Munoz, J Andrew Bagnell, and Martial Hebert. 3-d scene analysis via sequenced predictions over points and regions. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2609–2616. IEEE, 2011.

[337] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l 0 gradient minimization. In *Proceedings of the 2011 SIGGRAPH Asia conference*, pages 1–12, 2011.

[338] Yang Xu, Yuequan Bao, Jiahui Chen, Wangmeng Zuo, and Hui Li. Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images. *Structural Health Monitoring*, 18(3):653–674, 2019.

[339] Tomoyuki Yamaguchi and Shuji Hashimoto. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, 21(5):797–809, 2010.

[340] Seung-Nam Yu, Jae-Ho Jang, and Chang-Soo Han. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in construction*, 16(3):255–261, 2007.

[341] Christopher Zach, Christian Häne, and Marc Pollefeys. What is optimized in convex relaxations for multilabel problems: Connecting discrete and continuously inspired map inference. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):157–170, 2013.

[342] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*.

[343] Dengyong Zhou and Bernhard Schölkopf. Regularization on discrete spaces. In *Joint Pattern Recognition Symposium*, pages 361–368. Springer, 2005.

[344] Zhenhua Zhu and Ioannis Brilakis. Machine vision-based concrete surface quality assessment. *Journal of Construction Engineering and Management*, 136(2):210–218, 2010.