Imperial College London
Department of Computing

Optimisation for image processing

Luong Vu Ngoc Duy

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College and
the Diploma of Imperial College, 2014

Abstract

The main purpose of optimisation in image processing is to compensate for missing, corrupted image data, or to find good correspondences between input images. We note that image data essentially has infinite dimensionality that needs to be discretised at certain levels of resolution. Most image processing methods find a suboptimal solution, given the characteristics of the problem. While the general optimisation literature is vast, there does not seem to be an accepted universal method for all image problems. In this thesis, we consider three interrelated optimisation approaches to exploit problem structures of various relaxations to three common image processing problems:

1. The first approach to the image registration problem is based on the nonlinear programming model. Image registration is an ill-posed problem and suffers from many undesired local optima. In order to remove these unwanted solutions, certain regularisers or constraints are needed. In this thesis, prior knowledge of rigid structures of the images is included in the problem using linear and bilinear constraints. The aim is to match two images while maintaining the rigid structure of certain parts of the images. A sequential quadratic programming algorithm is used, employing dimensional reduction, to solve the resulting discretised constrained optimisation problem. We show that pre-processing of the constraints can reduce problem dimensionality. Experimental results demonstrate better performance of our proposed algorithm compare to the current methods.

2. The second approach is based on discrete Markov Random Fields (MRF). MRF has been successfully used in machine learning, artificial intelligence, image processing, including the image registration problem. In the discrete MRF model, the domain of the image problem is fixed (relaxed) to a certain range. Therefore, the optimal solution to the relaxed problem could be found in the predefined domain. The original discrete MRF is NP hard and relaxations are needed to obtain a suboptimal solution in polynomial time. One popular approach is the linear programming (LP) relaxation. However, the LP relaxation of MRF (LP-MRF) is excessively high dimensional and contains sophisticated constraints. Therefore, even one iteration of a standard LP solver (e.g. interior-point algorithm), may take too long to terminate. Dual decomposition technique has been used to formulate a convex-nondifferentiable dual LP-MRF that has geometrical advantages. This has led

to the development of first order methods that take into account the MRF structure. The methods considered in this thesis for solving the dual LP-MRF are the projected subgradient and mirror descent using nonlinear weighted distance functions. An analysis of the convergence properties of the method is provided, along with improved convergence rate estimates. The experiments on synthetic data and an image segmentation problem show promising results.

3. The third approach employs a hierarchy of problem's models for computing the search directions. The first two approaches are specialised methods for image problems at a certain level of discretisation. As input images are infinite-dimensional, all computational methods require their discretisation at some levels. Clearly, high resolution images carry more information but they lead to very large scale and ill-posed optimisation problems. By contrast, although low level discretisation suffers from the loss of information, it benefits from low computational cost. In addition, a coarser representation of a fine image problem could be treated as a relaxation to the problem, i.e. the coarse problem is less ill-conditioned. Therefore, propagating a solution of a good coarse approximation to the fine problem could potentially improve the fine level. With the aim of utilising low level information within the high level process, we propose a multilevel optimisation method to solve the convex composite optimisation problem. This problem consists of the minimisation of the sum of a smooth convex function and a simple non-smooth convex function. The method iterates between fine and coarse levels of discretisation in the sense that the search direction is computed using information from either the gradient or a solution of the coarse model. We show that the proposed algorithm is a contraction on the optimal solution and demonstrate excellent performance on experiments with image restoration problems.

Acknowledgements

Doing a PhD is like solving an optimisation problem. Sometimes, it does not going so well, like the problem gets stuck in a local optimum. Sometimes, many ideas have to be considered, many topics need to be studied, just like computing multiple search directions. Without the support and inspiration of my supervisors, my parents, colleagues and friends, this work would have never been accomplished, and the problem may never converge. First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Berc Rustem, for his constant guidance he has given me during my PhD both academically and personally. I would like to thank my second supervisor, Professor Daniel Rueckert for his support and encouragement throughout the course of this work. I also cannot overstate my appreciation to my co-supervisor, Dr. Panos Parpas for his invaluable help and motivations during the last two years of my PhD.

For their supports in various literatures in image processing and optimisation, I would like to thank Dr. Daniel Kuhn, Dr. George Tzallas-Regas, Dr. Wolfram Wiesemann, Dr. Paul Aljabar, Dr. Wenzhe Shi, Dr. Kanwal Bhatia, Dr. Luis Pizarro and Dr. Ankur Handa. I really enjoyed their valuable discussions on topics related to this thesis.

I am grateful to my friends and colleagues for amusing (or irritating) me during my study, Micheal Hadjiyiannis, Vladimir Roitch, Iakovos Kakouris, Trang Quang Kha, Chin Pang Ho, Loizos Markides, Dr. Ricardo Guerrero Moreno, Dr. Robin Wolz and Dr. Angelos Georghiou.

Special thanks to Chong Hay Wong, Hoang Dieu Anh, Nguyen Van Phinh, Le Duc Anh, Van Tien Hieu, and many others for making colourful and enjoyable periods in my life.

Last but not least, I want to thank my parents, Lam and Lua, lovely sister Linh and brother Dung for their loving support through all these years. Without them all I would not be where I am today.

My research was partially funded by the Engineering and Physical Sciences Research Council (EPSRC) via the EP/I014640/1 award for the ”Robust Optimization of Nonlinear Processes under Uncertainty” project.

To my parents, Dr. Luong Ngoc Lam and Ms. Vu Thi Lua,

to my siblings, Miss. Luong Vu Thi Ngoc Linh and Mr. Luong Vu Ngoc Dung

Declaration

This thesis presents my work in the Department of Computing at Imperial College London between October 2009 and March 2014. This work were done under the supervision of Prof. Berc Rustem, Prof. Daniel Rueckert and Dr. Panos Parpas. I declare that the work presented in this thesis is my own, except where acknowledged. This thesis has not been submitted for the award of any degree or diploma in any other tertiary institution.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The importance of image processing is due to the need for extracting as much information as possible from available image data. The rapid development in technology has resulted in increased volumes of image data. Processing this data requires increased computational effort. To maintain adequate performance, efficient techniques are needed for image processing. This has received substantial attention in recent decades. Many methods have been developed, based on optimisation algorithms and their efficient software implementation. The basic approach in each case is based on the explicit or implicit optimisation of a specified criterion. The aim is always to efficiently extract useful information from the images, such as patterns, noise and similarities.

This thesis focuses on gradient-based optimisation methods for image processing. Three interrelated optimisation approaches are considered to solve *three* common computer vision problems. Image problems are generally ill-conditioned and suffer from undesirable local optimal solutions. While there is a vast literature on the topic of optimisation methods, it is difficult to solve an image problem without taking into account prior knowledge, image structure, or its relaxation model. Thus for particular applications, specialised methods are essential to obtain meaningful image information. The methods considered in this thesis take into account three type of relaxations to ill-posed image processing problem: in Chapters 2 and 3, relaxation based on *prior knowledge* of image objects; in Chaters 4 and 5, convex relaxation via Markov Random Fields and *efficient solvers for optimisation subproblems*; and, in Chapter 6, *the hierarchy of image resolutions* where the coarser model is a smoother version of a fine and ill-posed problem. The final Chapter presents the thesis conclusions, and discusses some shortcomings of the proposed methods. Current and future directions that are directly related or have been motivated by some of the issues raised in this thesis are summarised.

1.1 Image Registration

Chapter 2 reviews the background of image registration, a popular procedure in various applications of computer vision, especially in medical imaging. Image registration is the process of aligning two or more images of the same scene. This process involves designating one image as the reference (also called the reference image or the fixed image), and applying geometric transformations to the other images (referred as templates or moving images) so that they align with the reference. Images can be misaligned for a variety of reasons. Commonly, the images are captured under variable conditions that can change camera perspective. Misalignment can also be the result of lens and sensor distortions or differences between capture devices.

Image registration is often used as a preliminary step in other image processing applications. For example, you can use image registration to align satellite images or to align medical images captured with different diagnostic modalities (MRI and SPECT). Image registration allows you to compare common features in different images. For example, you might discover how a river has migrated, how an area became flooded, or whether a tumor is visible in an MRI or SPECT image. In general, the process of image registration involves finding the optimal

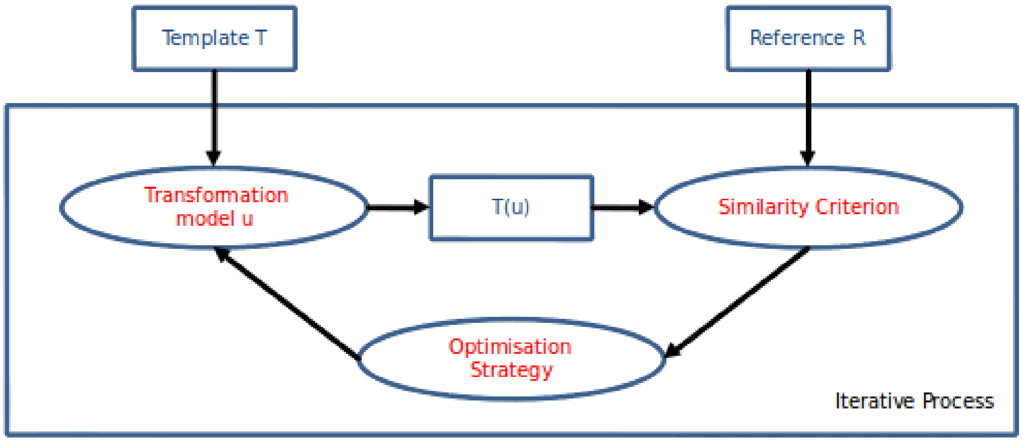


Figure 1.1: *Image registration framework*

geometric transformation which maximises the correspondences across the images. A geometric transformation maps locations in one image to new locations in another image. The step of determining the correct geometric transformation parameters is key to the image registration process. This involves several components (see Figure 1.1):

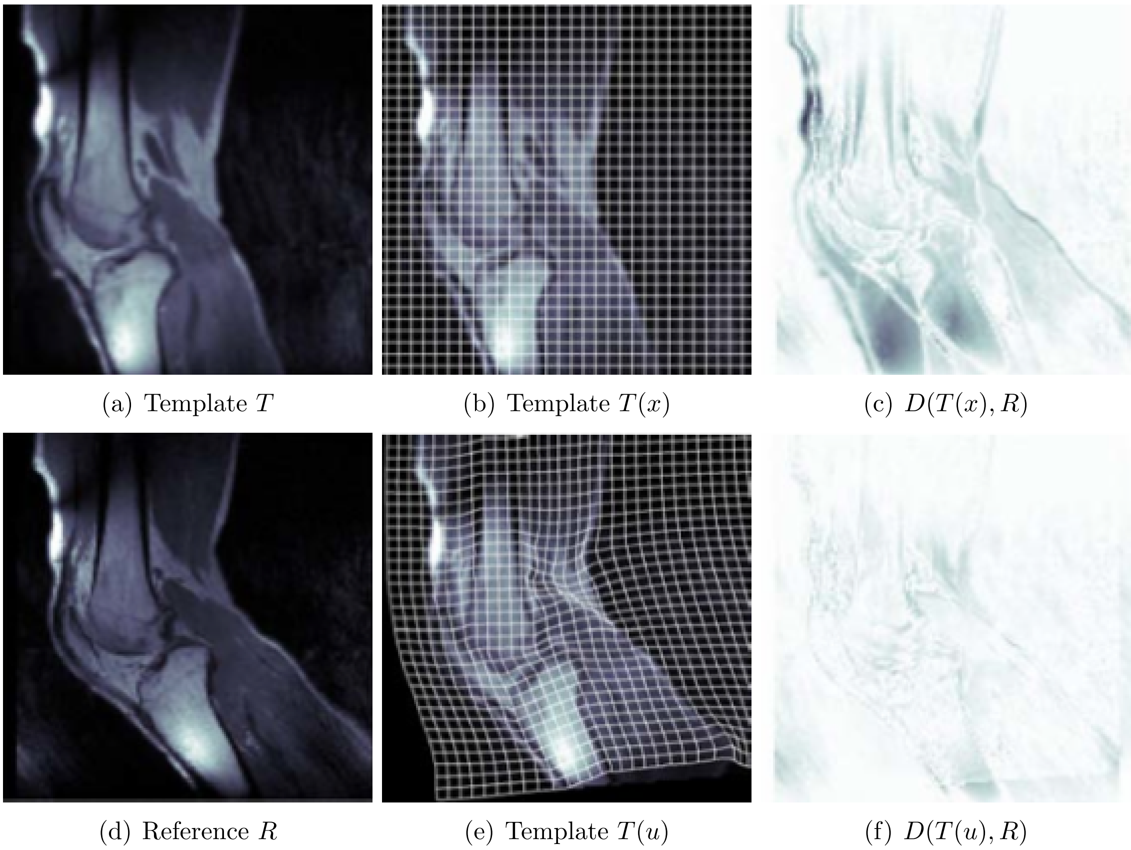
- **Transformation model** defines a geometric transformation between the images. In simple applications, such as linear, rigid, affine registration, the transformation can be defined by a matrix. In general automatic registration problems, non-rigid transformations are employed for improved accuracy of image alignment. In this case, for any initial location x in the image domain Ω, one needs to seek for a location mapping $u(x) : \Omega \rightarrow \Omega$ to apply on the template image. The image template T is indeed a constant image interpolation function [69], which compute image intensity given pixel locations, i.e. $T(u) : \Omega \rightarrow \mathbb{R}$.

- **Similarity criterion** measures the satisfaction of alignment between the images. In cases where features such as landmarks, edges or surfaces are available, the distances between corresponding features can be used to measure the alignment. In other cases the image intensities can be directly used to measure the alignment. Various mathematical models [69] can be used to define the distance between images in the form $D(T(u), R)$, where D denotes the distance function. Recent registration models incorporate the regularisation function $S(u)$ and constraints $C(u)$ in order to remove undesirable transformations and improve the registration quality.

- **Optimisation strategy** maximises the similarity criterion. Given that a transformation model and a similarity criterion have been defined, non-rigid registration can be formulated as an optimisation problem whose goal is to find the optimal geometric transformation to match the template to the reference:

$$\begin{aligned} \underset{u}{\text{minimise}} \quad & D(T(u), R) + S(u) \\ \text{subject to} \quad & C(u) = 0 \end{aligned}$$

Figure 1.2 shows a simple visual illustration of an image registration problem. In this example, we are given a reference image R and a template image T; and the task is to investigate the movement of the template pixels in order to align the template with the reference. Non-rigid transformation model is chosen for this example, where x represents initial location of



(a) Template T (b) Template $T(x)$ (c) $D(T(x), R)$

(d) Reference R (e) Template $T(u)$ (f) $D(T(u), R)$

Figure 1.2: *Visual illustration of an image registration problem*

template pixels and u is the desired location mapping, corresponding to the movement of pixels. The goal is to find a geometric transformation u to apply on T so that $T(u)$ should look as similar as possible to R. A similarity criterion $D(T, R)$ is designed by considering the intensity differences between image. Figure 1.2(b) and Figure 1.2(c) illustrate the location vector x and the dissimilarity measure prior to registration. Figure 1.2(e) and Figure 1.2(f) illustrate the location mapping u (the desired transformation) and the dissimilarity measure in the post-registration. In order to obtain the latter, a gradient descent method is employed in the optimisation strategy.

In some clinical applications, certain local characteristics of objects in the image should be preserved. For example, any deformation of human organs should preserve the volume of soft tissues. Another example is a transformation of a bony object is required to maintain its rigid structure. In this thesis, a registration problem with local rigidity constraints is examined, as it also includes the popular volume preservation requirement. The continuous problem formulation and the development of constraints are described for infinite dimensional input images.

There are two technical issues to address in solving this problem, discretisation *(for the transformation model)* and optimisation, and are discussed in Chapter 3. The initial input image data is infinite dimensional. Therefore, it needs to be discretised in order to transcribe it as a finite optimisation problem. Choosing a discretisation method is a delicate matter that can affect the performance of the infinite problem. Since local rigidity is only applicable to certain pixels that correspond to bony objects, while other parts of the image can deform freely, it is necessary to control the displacement of every pixel. We employ the staggered grid discretisation technique that enables the explicit imposition of the constraints on every pixel. The resulting finite optimisation problem has linear and bilinear constraints:

$$\begin{aligned} \underset{u}{\text{minimise}} \quad & F(u) \\ \text{subject to} \quad & Au = 0 \\ & C(u) = 0 \end{aligned}$$

A dimensional reduction technique is applied to obtain a reduced dimensional nonlinear programming problem:

$$\begin{aligned} \underset{u^z}{\text{minimise}} \quad & F(Zu^z) \\ \text{subject to} \quad & C(Zu^z) = 0 \end{aligned}$$

In Chapter 3, a sequential quadratic programming (SQP) algorithm is developed to solve the constrained registration problem. The effectiveness of the proposed method is demonstrated on both synthetic and real Magnetic Resonance image registration problems, with promising results. The method is also compared against the penalty method [70]. The latter requires a judicious choice of the penalty parameter in order to provide a balance between the constraints and the objective function. The proposed SQP method reduces problem dimensionality, avoids sensitivity to penalty parameter choice and guarantees good feasibility (rigid movement of bony structures). The rigid movement of bony structures are important because a meaningful rigid transformation should contiuously maintain the structure. This work has been published in 2011 [67]:

Luong, D.V.N., Rueckert, D., and Rustem, B. Incorporating hard constraints into non-rigid registration via nonlinear programming. *Proc. SPIE. Medical Imaging: Image Processing (2011).*

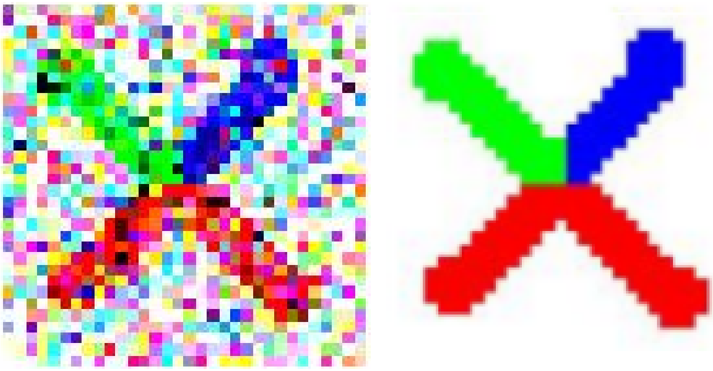
1.2 Markov Random Fields minimisation

SQP algorithm is developed to solve an ill-conditioned image registration problem that should not admit solutions violating rigid properties of certain parts in the image. The approach illustrates that adding constraints in image processing is one way to reduce the non-uniqueness of solutions to the problem. Another way to relax an image problem is using convex relaxation technique via Markov Random Fields (MRF) [65]. MRF is a popular framework in image and signal processing, machine learning and artificial intelligence. Recently, the registration problem has been sucessfully solved using MRF model [31]. In their study, a MRF model of image registration is formulated utilising the parametric transformation framework [90]. MRF is applicable to problems with finite dimensionality, for example image registration with a finite number of pixels or control points, and image segmentation with a finite number of pixels or features. In the recent years, MRF model and its corresponding optimisation problem have been considered in numerous studies (see [102] and references therein). This thesis considers the MRF minimisation approach based on the first order (gradient based) optimisation method.

In Chapter 4, MRF background and its application in image processing are described. MRF is originated from the probabilistic theory, however, as we shall see, MRF is equivalent to the multi-labelling problem on an undirected graph $G = (V, E)$, where V, E denote a set of nodes and a set of edges respectively. Multi-labelling on a graph is a very popular model in image processing. In general, most image processing problems aim to reveal some hidden quantities x based on some visual observations. Every hidden quantity corresponds to a feature, or a pixel, or an object of the observed image; and it belongs to a set of nodes V that made up the graph G. Each hidden quantity x_a, for all $a \in V$, can be assigned a value from a set of discrete labels L, where each label represents a feasible solution for the corresponding hidden quantity. Each label assignment is subject to a cost of labelling $\theta_a(x_a)$, which encodes how much the assignment of label $x_a \in L$ to node a disagrees with the observed image data at node a. Furthermore, the labelling at a node a also has influences on its neighbouring nodes. The term *neighbouring* defines the edges E of the graph G . The neighbouring influence is often known a priori, and encoded into the pairwise cost $\theta_{ab}(x_a, x_b)$. One way to achieve the optimal labelling for G is to minimise the cost of all possible combinations of hidden quantities and observed image data:

$$\underset{x_a \in L}{\text{minimise}} \sum_{a \in V} \theta_a(x_a) + \sum_{ab \in E} \theta_{ab}(x_a, x_b)$$

This optimisation problem is known to be NP-hard and has exessively large dimensionality. Therefore efficient optimisers are essential for this commonly used MRF model for solving image problems. A simple example of MRF model, or multilabelling prolem, is illustrated in Figure 1.3. In this example, each node a corresponds to an image pixel, whilst a pair of neighbouring pixels form an edge ab. Each node associates with a hidden quantity $x_a \in L$, where the label set L contains 4 possible colours {white, red, green, blue}. In addition, there are given unary cost $\theta_a(x_a)$ and pairwise cost $\theta_{ab}(x_a, x_b)$ for each label assignment for a node and for a pair of neighbouring nodes respectively. The cost function is designed in such a way that it is less expensive for a more likely label assignment. The objective of the multilabelling problem is to obtain a label assignment for all nodes such that the total cost is minimised. Chapter 4 presents a review of various MRF optimisation approaches based on dynamic programming and combinatorial optimisation. A large part of Chapter 4 focuses on the dual decomposition of the linear programming (LP) relaxation of the MRF model (LP-MRF). The final problem described in Chapter 4 turns out to be a nonmooth optimisation problem. Chapter 5 develops first order methods for the convex nonsmooth LP-MRF. First, the standard subgradient projection is considered. Subsequently, the model is reparameterised to obtain the augmented optimisation



(a) Corrupted image (b) Segmented image

Figure 1.3: *MRF model / Multilabelling for an image segmentation problem*

problem:

$$\begin{aligned}
\underset{\{\rho\in\Delta\},\{\lambda\in\Lambda\}}{\text{maximise}} \quad & \sum_{t\in T} \min_{x^t\in X^t} \langle \rho^t.\theta + \lambda^t, x^t \rangle \\
\text{where} \quad & \Delta = \left\{ \rho \,\middle|\, \forall i \in \mathcal{I} : \sum_t \rho_i^t = 1, \rho_i^t \geq 0 \right\} \\
& \Lambda = \left\{ \lambda \,\middle|\, \forall i \in \mathcal{I} : \sum_t \lambda_i^t = 0 \right\}
\end{aligned}$$

This leads to the development of a nonlinear weighted projection method to solve the augmented problem. The newly proposed projection is based on the theory of mirror descent [4, 48, 76]. The method employs proximal Bregman distance concepts, where a weighted Entropy distance and a weighted Euclidean distance for nonlinear projection is developed. Apart from the weighted distances, another novel development is the adoption of a weighted norm and weighted Lipschitz constant. The convergence properties of both methods are analysed, establishing the superiority of the proposed method. Furthermore, convergence analysis is performed to identify the optimal step-size strategy for the entropy projection and the adaptive step-size for euclidean projection. Some popular examples for MRF problems are used in the experiments to study the empirical performance of the weighted projection method (mirror descent) and the standard subgradient projection. This work has been published in Lecture Notes in Computer Science [66] and is under reviewed for a journal.

Luong, D.V.N., Parpas, P., Rueckert, D., and Rustem, B. Solving MRF minimization by Mirror Descent. In *Advances in Visual Computing*, Lecture Notes in Computer Science. 2012.

1.3 Multilevel Optimisation

The first two approaches are developed to solve finite optimisation problems in image processing. Clearly, these finite problems arise from a certain level of discretisation as the original image domain is infinite dimensional. A finer discretisation will have more information about the images but requires more computational resources. In Chapter 6, a new algorithm is developed to employ different levels of discretisation. The expectation is that, low cost steps, based on the coarse discretised problem, can be used within the high cost iterations of the finer discretisation, while maintaining the convergence properties of the original problem. At each iteration of the finely discretised level, the algorithm computes either direct search, using the gradient at current level, or a coarse correction. The coarse correction is generated from the solution of a surrogate optimisation problem at the lower (coarse) resolution. This correction step can significantly improve progress towards the optimal point as long as the coarse problem maintains the character of the fine problem. The relationship between levels of discretisation is defined by the first order coherence property in the sense that there is a local equivalence of the *gradient mapping* between levels of discretisation. This has a crucial role in the development of multilevel algorithms and their convergence. Based on the initial multilevel scheme [73] for unconstrained optimisation, a novel multilevel proximal method is developed for composite convex problems that consists of a convex Lipschitz-gradient function $f(x)$ and a simple nonsmooth function $g(x)$ with simple constraints Ω,

$$\underset{x\in\Omega}{\text{minimise}} \quad f(x) + g(x)$$

The above problem is very popular in image processing where $f(x)$ often represents matching criteria and $g(x)$ penalises unwanted solutions $x \in \Omega$. Composite convex optimisation has received a lot of attention in inverse problems and imaging in recent years. Its special case of defining $g(x)$ as a simple norm function has been used extensively to solve the following seminal model of image deterioration:

$$b = Ax + \epsilon$$

where $x \in \mathbb{R}^n$ is the true (unknown) image, $A : R^n \to R^m$ is a known image transformation, $\epsilon \in \mathbb{R}^m$ is a perturbation vector and $b \in \mathbb{R}^m$ is an observed input image. For instance, A can be an identity matrix, and the image model becomes a denoising problem. If A is constructed from a Point Spread Function [41], then the model becomes a deblurring and denoising problem (image restoration). When A is an irregular sampling and a convolution, the model represents a super-resolution problem [26]. In the case where A contains an image dictionary, then the model becomes classsification or recognition problems [115]. Other applications include image inpainting, compression noise reduction, texture and cartoon decomposition, ... Finding x from the observation b is an inverse problem. Due to various structures of A and the noise ϵ, closed-form solutions hardly exist. There are many specialised techniques to solve each particular problem. However, the following special formulation of composite convex optimisation is often considered as a promising candidate to solve for such an inverse problem:

$$\underset{x\in\Omega}{\text{minimise}} \quad \|Ax - b\|_2^2 + \|Wx\|_1$$

In this special composite convex optimisation model, the first term is a smooth convex Lipschitz function, the second term is a simple nonsmooth convex function. The additional linear operator W defines a type of regularisation to impose on the solution. For example, if W is a discrete gradient, then it corresponds to the total variation regularisation [5]. In image processing, the total variation penalty is popular for its ability to maintain sharp edges. When W is a type of wavelet transform, it is equivalent to a Besov semi-norm [22]. The underlying philosophy in using the latter regularisation is that most images have a sparse representation in the wavelet domain, and l_1 norm promotes sparsity. The aim of the composite convex problem is to look for an image x which minimises $\|Wx\|_1$ such that the transformation Ax is close to b.

In Chapter 6, we show various techniques to construct a hierarchy of general composite convex models, and establish the relationship between them. A novel multilevel method is then proposed and its global convergence is proved. We demonstrate the effectiveness of our methods by excellent performance for the image restoration problem. This work has been submitted for publication. The paper and associated code can be found at:

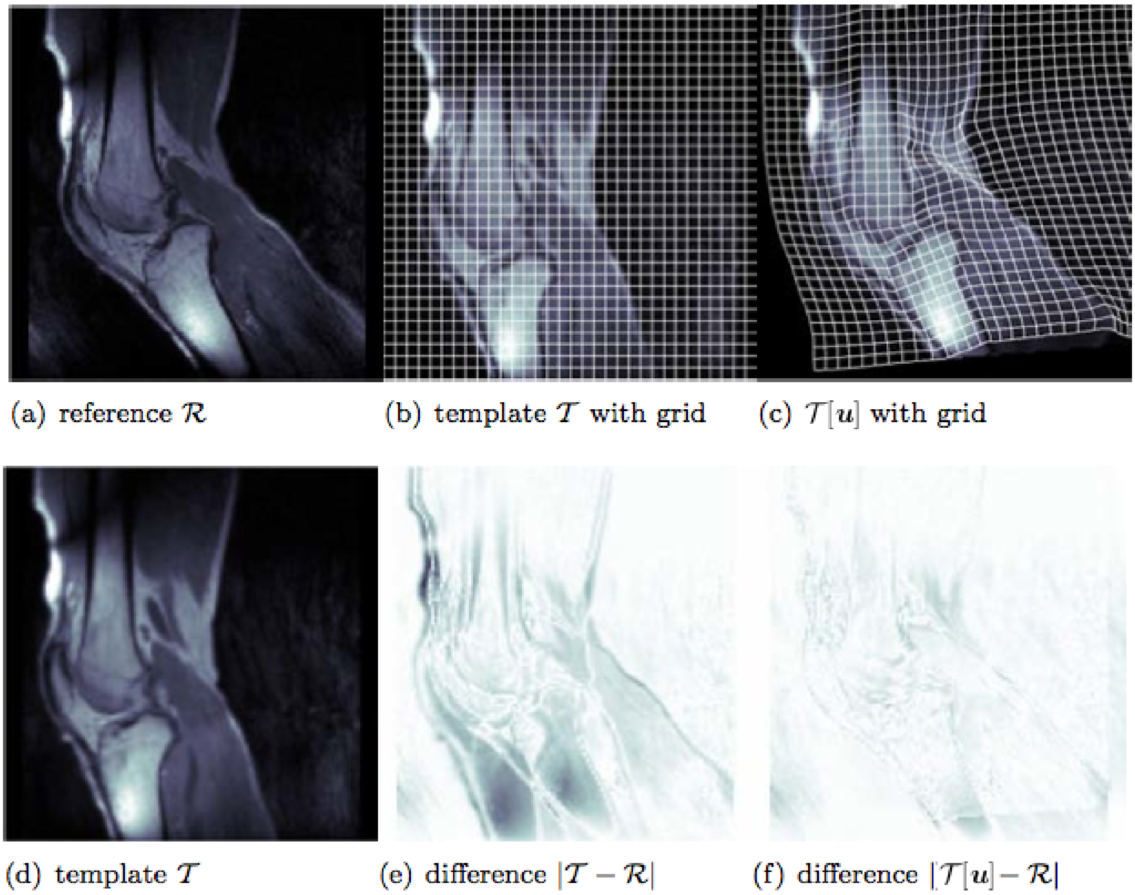
http://www.doc.ic.ac.uk/ pp500/mista.html

Chapter 2

Image registration

Image registration is a fundamental task in image processing in general and in medical imaging in particular[39]. Given two images taken at different times, on different devices or from different perspectives, the goal is to determine a reasonable spatial transformation, such that a transformed version of one image is similar to the other image. A simple registration is illustrated in Figure 2.1: (a) and (b) show slices of Magnetic Resonance scans of a human knee taken at two different times. The objective is to find a transformation grid (c) that minimise the dissimilarities due to the different positions of the knee. From the example, we can see an improvement in the alignment of the images when the transformation u is applied on the template image $\mathcal{T}$, i.e. there is less difference in the post-registration ($|\mathcal{T}[u] - \mathcal{R}|$) compared to pre-registration ($|\mathcal{T} - \mathcal{R}|$).

Image registration has become an essential tool in many scientific topics, including biology, chemistry, criminology, genetics, and medicine. More specific examples include remote sensing, motion correction, verification of pre- and post-intervention images, and the study of temporal series of cardiac images. Image acquisition techniques, such as computer tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), single-photon emission computer tomography (SPECT), or ultrasound, have been the subject of significant development in recent years. This in turn has led to a remarkable increase in the demand for visuallisation and analysis techniques for clinical applications.



(a) reference $\mathcal{R}$ (b) template $\mathcal{T}$ with grid (c) $\mathcal{T}[\boldsymbol{u}]$ with grid

(d) template $\mathcal{T}$ (e) difference $|\mathcal{T} - \mathcal{R}|$ (f) difference $|\mathcal{T}[\boldsymbol{u}] - \mathcal{R}|$

Figure 2.1: *Modified slices from CT scans of human knees; imaged by Thomas Netsch, Philips Research Hamburg.*

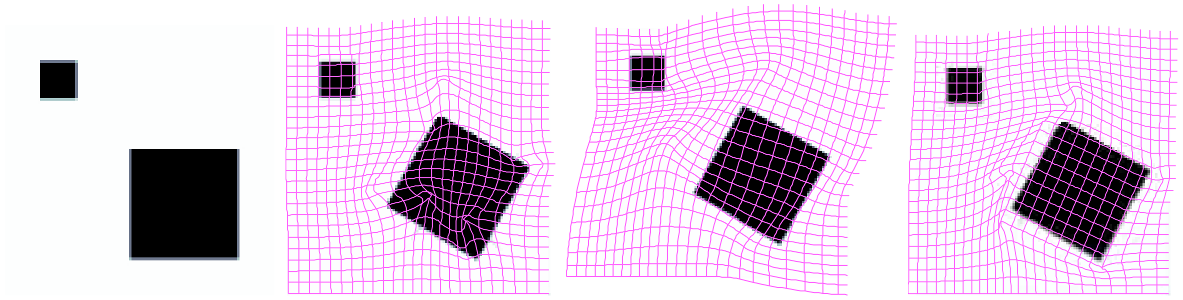
Optimisation is a key component of any image registration framework. Most registration methods can be explained from an optimisation point of view, implicitly or explicitly. In general, image registration problems are nonconvex and often solved by gradient-based optimisation methods. Rueckert et al. [90] introduce a well-known parametric model based on a B-spline basis function and use a gradient descent method to solve the problem. Pennec et al. [85] establish the relationship between vector-force algorithms and gradient descent methods. Vercauteren et al. [108] provide theoretical justifications on the equivalence between vector-force variants and various gradient based optimisers. In addition, Haber and Modersitzki [38, 69] develop a general variational framework that can tackle the registration problem from a complete optimisation perspective. In their framework, hierarchical resolutions, with a sequence of coarse to fine discretised images, are employed in the solver. The method has two advantages: firstly, the low resolution image problem benefits from low computational cost; secondly, a solution of a coarse problem provides a good initial starting point for a finer problem. While the use of multi-resolution optimisation is very popular in image registration, it is often used without principled justifications. A more rigorous theory of multi-resolution is presented in Chapter 6 in terms of multilevel methods.

Unlike nonconvex optimisations, there have been several studies on convex optimisation methods for the registration problem. Taylor et al. [104] introduce a convex approximation of the original objective function. Their method restricts the displacements of grid points to a fixed range and utilises the interior point method to solve the linear programming problem. Glocker et al. [31] reformulate the parametric registration problem as a Markov Random Field (MRF) minimisation problem. They associate each displacement with a discrete label, and construct a discrete multi-labelling problem. The discrete problem belongs to the popular MRF model in computer vision. Although the method produces a globally optimal solution, the solution is only optimal within the defined discrete search space. Effectively, fine local displacements are not covered.

In the first approach, we consider the image registration problem with local rigid constraints to penalise undesirable solutions where rigid structures of the image are not preserved. This requires the explicit control of fine, local displacements. To address this, a nonlinear programming method is developed to solve the nonconvex registration problem in the continuous search space.

2.1 Constrained registration

Nonrigid registration is a common technique in medical image processing. The registration problem is ill-posed: several possible transformations exist that will result in perfect image alignment. One method to approach this is to add a regulariser, which penalises non-smooth solutions. However, the problem may still remain ill-posed. More recent approaches incorporate additional prior knowledge, for instance, regularisers such as elastic, curvature, folding constraints or those such as volume preservation and local rigidity. This additional information reduces the level of non-uniqueness significantly and therefore produces a more realistic, meaningful transformation. Figure 2.2 illustrates an example of registration results using prior information. All transformations lead to the same target image and the transformed images look similar visually. However the registration in (d) with local rigid constraints produces a



(a) Original Image (b) Elastic Regulariser (c) Curvature Regulariser (d) Elastic + Rigid Constraints

Figure 2.2: *Registration with prior knowledges: Elastic, Curvature, Elastic and Local Rigid Constraints.*

more plausible transformation since it locally preserves the internal structure of the original image.

Methods to solve constrained registration are important in medical imaging, due to the need for reliable and biologically meaningful transformations. In [88], Rohlfing et al. apply volume preservation constraints to the B-spline model framework. They use a mutual information cost function with additional penalty terms for incompressibility (using the absolute value of the log of the Jacobian of the transformation, computed by finite differences), and smoothness (computed by the second order derivatives of the deformation). The model is posed as an unconstrained minimisation problem. Staring et al. [99] add constraints to penalise the cost function to maintain the structure of rigid objects in the B-spline framework. These constraints include: linearity, orthogonality and volume preservation. They evaluate the rigidity constraints over the control points, which can be expensive when many control points needed for small rigid regions.

Modersitzki [70] incorporates rigidity constraints into his variational framework. The constrained terms are similar to Staring et al. [99] and are also used to penalise the cost function. He computes the penalty on all pixels and uses a weighted differentiable function. The method has been shown to work well-chosen values for the penalty parameters, but the need for such a choice is a disadvantage of the technique. Without prior experience or extensive testing, it is difficult to select an appropriate penalty parameter. In addition, since the constraints are penalised in the cost function, there is no guarantee of producing a feasible solution. In the contrast, Haber and Modersitzki [37] propose a SQP method to solve the incompressible registration problem which maintains the feasibility of the solution.

This chapter studies registration with local rigidity constraints using the variational framework [70]. The model is general enough to utilise a large class of optimisers and it controls rigid characteristic of every pixel explicitly. Unlike volume preservation, which only has quadratic constraints, rigidity requires both linear constraints and quadratic constraints. Thus, QR decomposition is employed, in order to seek a solution in the null space of the linear constraints.

2.2 Mathematical model

The method considered here is strongly connected with variational methods used to solve the continuous registration problem [108]. The information obtained from image data is modelled in a continuous setting. This includes the distance function $\mathcal{D}(.,.)$ to quantify the dissimilaritiy between images, the regulariser $\mathcal{S}(.)$ to smooth the transformation and the constraints $\mathcal{C}(.)$ to enforce meaningful solutions. Let Ω be the image domain, i.e. feasible locations of an image pixel. Let $x \in \Omega$ be the initial location of every image pixel and x is unchanged. In 2D image, a pixel location consists of 2 components $x = (x^1, x^2)$, in 3D image, a voxel location consists of 3 components $x = (x^1, x^2, x^3)$. In this thesis, we consider 2D image problems, however, it is straightforward to extend for 3D images. For every pixel with initial location $x \in \Omega$, there is a corresponding location mapping $u \equiv u(x) : \Omega \to \Omega$. The problem is formulated as a constrained minimisation problem to find a new location u:

$$\begin{aligned} &\text{minimise} && \mathcal{D}\left(\mathcal{T}(u), \mathcal{R}\right) + \alpha \mathcal{S}(u) \\ &\text{subject to} && \mathcal{C}(u) = 0 \end{aligned} \tag{2.1}$$

The template image function $\mathcal{T}(u) : \Omega \to \mathbb{R}$ evaluates pixel intensity for every feasible location. For infeasible location, i.e. $\forall u \notin \Omega$, we set $\mathcal{T}(u) = 0$. The reference image $\mathcal{R} \equiv \mathcal{R}(x)$ is unchanged and holds intensity values for every pixels. In addition, α is a constant to balance the relative importance of the dissimilarity $\mathcal{D}(.,.)$ and the smoothness $\mathcal{S}(.)$. In the continuous setting, we have an infinite number of image pixels (with an infinite number of locations x) in the domain Ω.

Distance function is defined as the *sum-of-square* differences between the intensities of image pixels:

$$\mathcal{D}\left(\mathcal{T}(u), \mathcal{R}\right) = \frac{1}{2}\int_{\Omega} \left(\mathcal{T}(u(x)) - \mathcal{R}(x)\right)^2 \mathrm{d}x$$

Regulariser is chosen based on the application requirements. In the following sections, three basic regularisers are described: diffusion, elastic and curvature. Each regulariser can be reformulated to the form of a Euclidean norm of a partial derivative operator:

$$\mathcal{S}(u) = \frac{1}{2}\int_{\Omega} \|\mathcal{B}(\mathrm{d}u(x))\|^2 \mathrm{d}x \tag{2.2}$$

where $\mathrm{d}u(x) = u(x) - x$ holds feasible spatial *displacements*. For the ease of presentation, we use $u, \mathrm{d}u$ instead of $u(x), \mathrm{d}u(x)$ in the rest of this chapter and Chapter 3. In 2D image problem, $\mathrm{d}u = (\mathrm{d}u^1, \mathrm{d}u^2)$ represents displacements in horizontal and vertical directions.

- *Diffusion* regulariser damps the displacement between neighbourhood pixels. It was first proposed for optical flow and was the main ideas of Thirion's demon algorithm [106]:

$$\mathcal{S}^{Diff} = \frac{1}{2}\int_{\Omega} \left((\partial_1 \mathrm{d}u^1)^2 + (\partial_2 \mathrm{d}u^1)^2 + (\partial_1 \mathrm{d}u^2)^2 + (\partial_2 \mathrm{d}u^2)^2\right) \mathrm{d}x$$

It can be written in form of (2.2) with a diffusion operator $\mathcal{B}$ defined by a collection of partial derivatives about $\mathrm{d}u$:

$$\mathcal{B}^{Diff} = \begin{bmatrix} \nabla & 0 \\ 0 & \nabla \end{bmatrix} = \begin{bmatrix} \partial_1 & 0 \\ \partial_2 & 0 \\ 0 & \partial_1 \\ 0 & \partial_2 \end{bmatrix}$$

- *Curvature* regulariser penalises oscillations and implicitly allows affine transformation [27]:

$$\mathcal{S}^{Curv} = \frac{1}{2}\int_{\Omega} \left((\Delta \mathrm{d}u^1)^2 + (\Delta \mathrm{d}u^2)^2\right) \mathrm{d}x$$

where $\Delta v = \partial_{1,1} v + \partial_{2,2} v$. The curvature regulariser can be written in a compact norm penalty with partial derivative operator:

$$\mathcal{B} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix}$$

- *Elastic* regulariser is adapted to the elastic potential measuring the engergy when an elastic material is deformed. It was first used in image registration by Broit [21]. Let the divergence $\nabla \cdot \mathrm{d}u = \partial_1 \mathrm{d}u^1 + \partial_2 \mathrm{d}u^2$, then the elastic regulariser is formulated as:

$$\mathcal{S}^{Elas} = \frac{1}{2} \int_{\Omega} \left(\mu \|\nabla \mathrm{d}u\|^2 + (\lambda + \mu)(\nabla \cdot \mathrm{d}u)^2 \right) \mathrm{d}x$$

where λ and μ are Lamé constants [69]. The partial derivative operator is then given by:

$$\mathcal{B} = \begin{bmatrix} \sqrt{\mu}\,\nabla & 0 \\ 0 & \sqrt{\mu}\,\nabla \\ \sqrt{\lambda+\mu}\,\partial_1 & \sqrt{\lambda+\mu}\,\partial_2 \end{bmatrix}$$

Constraints Assuming rigid objects have been identified, constraints can be explicitly applied on every rigid pixel to allow for only rotation and translation. Pixels outside these regions are free to deform non-rigidly. The rigid movement of pixels is guaranteed by local neighbourhood displacements, which allows for only rotation, translation and volume preservation in the transformation.

Definition 2.2.1 *A transformation $u(x) \in C^2(\Omega, \mathbb{R}^2)$, i.e. $u(x) \in \Omega$ is a two dimensional vector and twice-continuously differentiable, is rigid if and only if there exist a rotation matrix R and a translation vector t such that $u(x) = Rx + t$.*

Lemma 2.2.2 *A transformation $u(x) \in C^2(\Omega, \mathbb{R}^2)$ is rigid if the following properties are satisfied:*

- *Linearity:* $\partial_{i,j} u^k = 0 \quad ; \quad i, j, k = 1, 2; \quad i \leq j$

$$\partial_{1,1}u^1 = 0, \partial_{1,2}u^1 = 0, \partial_{2,2}u^1 = 0; \; \partial_{1,1}u^2 = 0, \partial_{1,2}u^2 = 0, \partial_{2,2}u^2 = 0 \tag{2.3a}$$

- *Orthogonality:* $\nabla u \nabla u^\top = I$, *i.e.*

$$\begin{aligned} \partial_1 u^1 \odot \partial_1 u^1 + \partial_2 u^1 \odot \partial_2 u^1 &= e \\ \partial_1 u^1 \odot \partial_1 u^2 + \partial_2 u^1 \odot \partial_2 u^2 &= 0 \\ \partial_1 u^2 \odot \partial_1 u^2 + \partial_2 u^2 \odot \partial_2 u^2 &= e \end{aligned} \tag{2.3b}$$

- *Volume Preserving:* $\det(\nabla u) = 1$, *i.e.*

$$\partial_1 u^1 \odot \partial_2 u^2 - \partial_2 u^1 \odot \partial_1 u^2 = e \tag{2.3c}$$

where $e = (1, ..., 1)^\top$ *and* $\odot$ *denotes the Hadamard - pointwise product, i.e.* $x \odot y = (x_1 y_1, ..., x_n y_n)^\top$

Proof By definition 2.2.1, a rigid transformation can be written as $u(x) = Rx + t$ for some rotation matrix R and translation matrix t. Taking the second order derivative gives $\nabla^2 u(x) = 0$ which returns condition (2.3a). Since $\nabla u(x) = R$, and R is a rotation matrix, i.e. R is orthogonal, it follows simply that $RR^\top = I$ and $\det(R) = 1$, which are precisely the properties (2.3b) and (2.3c). ∎

The constraints in Lemma 2.2.2 can be collected in a nonlinear function, for $i, j, k = 1, 2; i \leq j$:

$$\mathcal{C}(u) = \begin{bmatrix} \partial_{i,j} u^k \\ \nabla u \nabla u^\top - I \\ \det(\nabla u) - 1 \end{bmatrix} = 0$$

yields the continuous model (2.1).

Discretisation of the continuous model (2.1) is essential because a continuous image contains infinite number of pixels. Some finite number of image pixels must be chosen in a discrete setting. There are two approaches for discretisation: *optimise-then-discretise* or *discretise-then-optimise*. In this work, the latter approach is chosen. To justify this, consider the optimality condition of (2.1) where an *elastic* regulariser is used:

$$
\begin{aligned}
0 \;=\;& \nabla_{u;v}\,\mathcal{D}\left(\mathcal{T}(u),\mathcal{R}\right) + \alpha\,\nabla_{u;v}\,\mathcal{S}(u) + \lambda^{\top}\,\nabla_{u;v}\,\mathcal{C}(u) \\
=\;& \int_{\Omega} \Big\langle \big(\mathcal{T}(u(x)) - \mathcal{R}(x)\big)\nabla\mathcal{T}(u(x)), v(x) \Big\rangle_{\Omega} \mathrm{d}x \\
& +\alpha \int_{\Omega} \left\langle (\lambda+\mu)(\nabla\cdot)^{\top}\nabla\cdot\mathrm{d}u(x) + \mu\Delta\mathrm{d}u(x), v(x) \right\rangle_{\Omega} \mathrm{d}x \\
& +\lambda^{\top}\,\nabla_{u;v}\,\mathcal{C}(u)
\end{aligned}
\tag{2.4}
$$

where $\nabla_{u;v}$ is Gâteaux derivative with respect to some pertubation $v(x)$ [69]. We skip the derivative of constraints due to its tedious formulation which includes third order on the transformation. Literature on the *optimise-then-discretise* technique [85, 108] discretise the Euler-Lagrange equation (2.4) *without* constraints. The discretisation must mimic the continuous setting and difficulties lie in computing complicated partial derivatives of displacements. In our system, the derivates of constraints go even beyond the second order, therefore the *discretise-then-optimise* technique is chosen to avoid error on discretisation of higher-order derivatives.

Choosing a discretisation method with mixed derivatives is a delicate matter. It is natural to use finite differences in most applications involving a differential operator. However, in this case, using finite central differences will lead to large errors resulting from the complication of partial derivatives in the constraints. In this thesis, a *discretise-then-optimise* technique is chosen. In particular, a staggered grid discretisation scheme is used. This has previously been succesfully applied to the stable discretisation of fluid flow and electromagnetics [36, 42], where operators such as the gradient, curl and divergence are involved.

The next chapter describes the staggered grid discretisation scheme and develops a novel optimisation framework to solve the discretised constrained registration problem.

Chapter 3

Discretisation and Optimisation

The continuous constrained registration problem was set up in the previous chapter as:

$$\begin{aligned} &\text{minimise} \quad && \mathcal{D}\left(\mathcal{T}(u), \mathcal{R}\right) + \alpha \mathcal{S}(u) \\ &\text{subject to} \quad && \mathcal{C}(u) = 0 \end{aligned}$$

As already mentioned in the previous chapter, a discretisation is required for the continuous model. In a discrete model, the image contains a finite number of image pixels with pixel width h. Clearly, varying pixel width h will change the number of pixels, i.e. image resolution. Denoting the width and height of the discrete image as n_h^1 and n_h^2, the problem dimension becomes $n_h = n_h^1 \times n_h^2$. For any discrete resolution h, we have a finite number of image locations $u \overset{\text{def}}{=} u_h(x) = (u_h^1(x), u_h^2(x)) \in C^2(\Omega_h, \mathbb{R}^2)$. The discretised version of the continuous model takes the finite number of pixels in the distance, regulariser and constrained functions, and approximates nonlinear partial derivatives by linear differential operators. The discretised optimisation problem is given in a general form (3.1) and is discussed in details in Section 3.1

$$\begin{aligned} &\underset{u}{\text{minimise}} \quad && D\left(T(u), R\right) + \alpha S(u) \\ &\text{subject to} \quad && C(u) = 0 \end{aligned} \tag{3.1}$$

3.1 Discretisation

3.1.1 Discretising u

We use staggered grid to discretise u at a certain image resolution h. At a resolution h, the images have $n := n_h = n_h^1 \times n_h^2$ pixels, each pixel being a squared-cell of lengths h. Normally, in image processing, pixels are identified by cell-centred grid points. However, in the staggered grid approach, we identify them by cell-edges. Given every pixel (i, j) with cell-centred $x_{i+\frac{1}{2}h,j+\frac{1}{2}h}$, $u(x)$ is discretised by its edges where $u^1(x) = \{x_{i,j+\frac{1}{2}h}, x_{i+h,j+\frac{1}{2}h}\}$ and $u^2(x) = \{x_{i+\frac{1}{2}h,j}, x_{i+\frac{1}{2}h,j+h}\}$, see Figure 3.1.

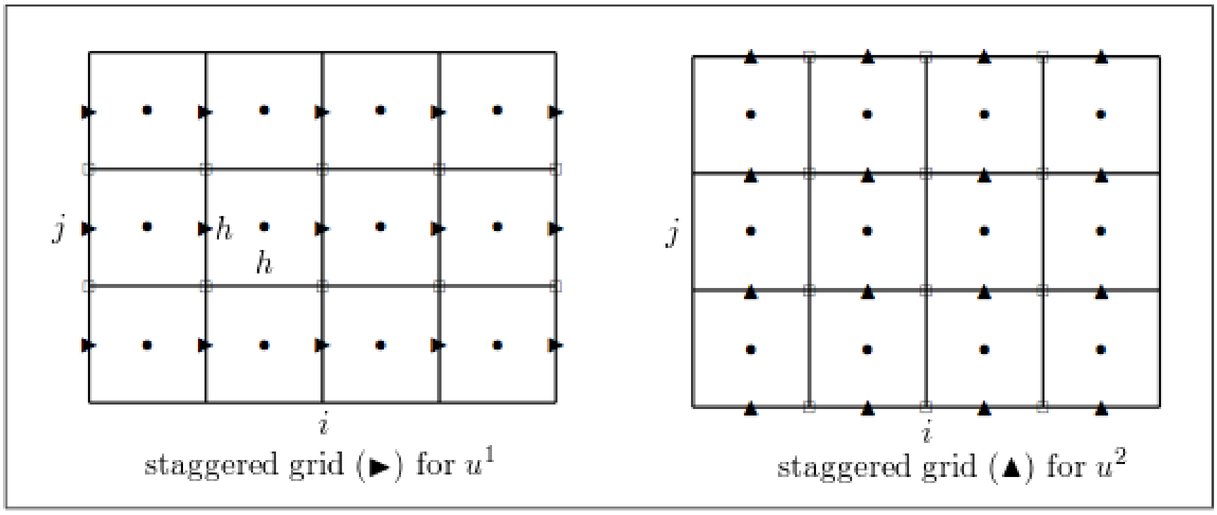


Figure 3.1: *Staggered grid for* 4×3 *image: cell-centred* $\bullet$*, cell-corner* $\square$*, cell-edges* $[u^1(\blacktriangleright), u^2(\blacktriangle)]$

The registration problem now aims to find a desired transformation (new location) of the staggered grid $u \overset{\text{def}}{=} (u^1(\blacktriangleright), u^2(\blacktriangle))$. Upon obtaining a staggered grid transformation, it can be converted to cell-centred grid by a shifting function (3.2). A shifting to cell-centre is required by the image interpolation function $T(u)$ to obtain a pixel intensity values based on the cell-centred position of the pixel.

Let band$(a_{-k}, ..., a_k; k_1, k_2)$ denote a $k_1 \times k_2$ matrix with diagonal bands $a_{-k}, ..., a_k$ where a_0 is on the main diagonal and $\otimes$ is the Kronecker product. For example:

$$\text{band}(1,2,3;3,4) = \begin{pmatrix} 2 & 3 & 0 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

then, let $P_j = \text{band}(0,1,1;n_j,n_j+1)/2 \otimes I_{n_i n_i}$ with $i,j = 1,2; i \neq j$. The shifting function is defined as:

$$P = \begin{pmatrix} P_1 & 0 \\ 0 & P_2 \end{pmatrix} \tag{3.2}$$

hence, a shifting to cell-centred grid, see Figure 3.1, is given by:

$$u(\bullet) = Pu = P \begin{bmatrix} u^1(\blacktriangleright) \\ u^2(\blacktriangle) \end{bmatrix}$$

3.1.2 Discretising $\mathcal{C}(u)$

Discretising the constraints $\mathcal{C}(u)$ requires discretised versions of first order derivatives ∂_i and second order derivatives $\partial_{i,j}$. An appropriate discretisation of the constraints is enough to cover all types of regularisers, therefore it is described here for a basis of functions containing partial derivatives. We use finite-differences between cell-edges to approximate the derivatives at the cell-centre of a pixel. By doing this, there is explicit control of the charateristics of the pixel. For each nonlinear partial derivatives $\partial_i, \partial_{i,j}$, linear discretised differential operators $\partial_i^k, \partial_{i,j}^k$ are defined, where k identifies a vector u^k which the operators apply to. For example, $\partial_{i,j}^1$ only applies to $u^1(\blacktriangleright)$. It is essential to apply the correct differential operators to mimic the continuous setting: $\partial_{i,j}^k u^k$.

Since all partial derivatives are defined at the centre of the pixel in the continuous setting, it is also necessary to define all discreted differential operators at the cell-centred $u(\bullet)$. Consider first-order derivatives ∂_i^1 on the staggered grid $u^1(\blacktriangleright)$: The normal direction (*direction* $\blacktriangleright$) derivatives $\partial_1^1 u^1(\blacktriangleright)$ are straightforward to compute, because their short difference lies on the cell-centred position. The tangential (*direction* $\blacktriangle$) short differences, ∂_2^1 of $u^1(\blacktriangleright)$, are located on the cell-corners $\square$ of the square pixel. Therefore, we approximate $\partial_2^1 u^1(\blacktriangleright)$ at the cell-centre $u(\bullet)$ by averaging its short differences on the 4 cell-corners $\square$, as shown in Figure 3.2. Similarly, the same arguments can be applied to $\partial_i^2 u^2(\blacktriangle)$.

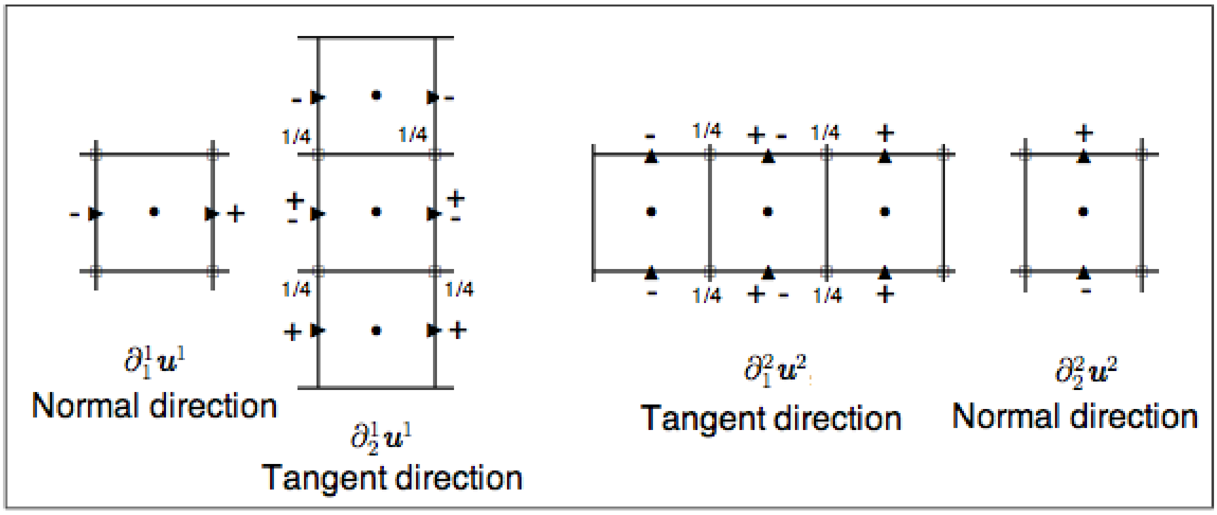

$\partial_1^1 \boldsymbol{u}^1$
Normal direction
$\partial_2^1 \boldsymbol{u}^1$
Tangent direction
$\partial_1^2 \boldsymbol{u}^2$
Tangent direction
$\partial_2^2 \boldsymbol{u}^2$
Normal direction
1/4
+
-

Figure 3.2: *Normal and tangential direction derivatives.*

Second-order derivatives are more complicated to illustrate graphically. However, one can think of them as taking normal and tangential short differences again on the definitions of first order derivatives. Explicit formulations of linear discretised derivatives are given by:

$$\begin{array}{rcl rcl}
\partial_1^1 & = & I_{n_2} \otimes D_1^l & \partial_1^2 & \underset{BC}{=} & P_2 \otimes D_1^s \\
\partial_2^1 & \underset{BC}{=} & D_2^s \otimes P_1 & \partial_2^2 & = & D_2^l \otimes I_{n_1} \\
\partial_{1,1}^1 & \underset{BC}{=} & I_{n_2} \otimes D_1^{l2} & \partial_{1,1}^2 & \underset{BC}{=} & P_2 \otimes D_1^{s2} \\
\partial_{1,2}^1 & \underset{BC}{=} & D_2^s \otimes D_1^l & \partial_{1,2}^2 & \underset{BC}{=} & D_2^l \otimes D_1^s \\
\partial_{2,2}^1 & \underset{BC}{=} & D_2^{s2} \otimes P_1 & \partial_{2,2}^2 & \underset{BC}{=} & D_2^{l2} \otimes I_{n_1}
\end{array} \tag{3.3}$$

where $\underset{BC}{=}$ subjects to Neumann boundary condition where necessary and the supported band diagonal matrices are defined as follows:

$$\begin{aligned}
P_j &= \text{band}(0, 1, 1; n_j, n_j + 1)/2 \\
D_j^s &= \text{band}(-1, 0, 1; n_j, n_j)/(2h_j) \\
D_j^l &= \text{band}(0, -1, 1; n_j, n_j + 1)/(h_j) \\
D_j^{s2} &= \text{band}(1, -2, 1; n_j, n_j)/(h_j^2) \\
D_j^{l2} &= \text{band}(0, 1, -1, -1, 1; n_j, n_j + 1)/(2h_j^2)
\end{aligned}$$

The defined differential operators (3.3) contain very few non-zero elements on each row (only non-zeros are at the *neighbours* involved in the normal or tangential differences). The number of rows is equivalent to the number of pixels in the image, thus the size of the differential operators are very large. However, not every pixel of the image is rigid, therefore the rows that correspond to non-rigid pixels can be removed from the operators (3.3). The continuous nonlinear constraints in Lemma 2.2.2 can be approximated in a discretised version as:

$$C(u) = \begin{cases} Au & = 0 \\ C_1^1 u \odot C_1^1 u + C_2^1 u \odot C_2^1 u - e & = 0 \\ C_1^1 u \odot C_1^2 u + C_2^1 u \odot C_2^2 u & = 0 \\ C_1^2 u \odot C_1^2 u + C_2^2 u \odot C_2^2 u - e & = 0 \\ C_1^1 u \odot C_2^2 u - C_2^1 u \odot C_1^2 u - e & = 0 \end{cases} \tag{3.4}$$

where A and C_j^i are collections of linear differential operators (3.3). These differential operators have been preprocessed to remove the rows associated with non-rigid pixels.

$$A = \begin{bmatrix} \partial_{1,1}^1 & 0 \\ \partial_{1,2}^1 & 0 \\ \partial_{2,2}^1 & 0 \\ 0 & \partial_{1,1}^2 \\ 0 & \partial_{1,2}^2 \\ 0 & \partial_{2,2}^2 \end{bmatrix} \quad C_1^1 = \left[\partial_1^1 \, , \, 0\right] \quad C_2^1 = [\partial_2^1 \, , \, 0] \quad C_1^2 = [0 \, , \, \partial_1^2] \quad C_2^2 = [0 \, , \, \partial_2^2] \tag{3.5}$$

Note that, the dimensionality of A can be furtherly reduced by removing linearly dependent rows via QR decomposition [81]. The Jacobian of $C(u)$ given by:

$$\nabla C(u) = \begin{bmatrix} A \\ 2\text{diag}(C_1^1 u) C_1^1 + 2\text{diag}(C_2^1 u) C_2^1 \\ \text{diag}(C_1^2 u) C_1^1 + \text{diag}(C_2^2 u) C_2^1 + \text{diag}(C_1^1 u) C_1^2 + \text{diag}(C_2^1 u) C_2^2 \\ 2\text{diag}(C_1^2 u) C_1^2 + 2\text{diag}(C_2^2 u) C_2^2 \\ \text{diag}(C_2^2 u) C_1^1 - \text{diag}(C_1^2 u) C_2^1 + \text{diag}(C_1^1 u) C_2^2 - \text{diag}(C_2^1 u) C_1^2 \end{bmatrix} \tag{3.6}$$

The second-derivative of $C(u)$ is used in conjunction with *Lagrange multipliers* λ when computing the Hessian (Section 3.2). For completeness, it is defined by:

$$\nabla^2 C(\lambda) = H_2^C(\lambda_2) + H_3^C(\lambda_3) + H_4^C(\lambda_4) + H_5^C(\lambda_5) \tag{3.7}$$

where $\lambda = [\lambda_2; \lambda_3; \lambda_4; \lambda_5]$ and:

$$\begin{aligned}
H_2^C(\lambda_2) &= 2C_1^1{}^\top \mathrm{diag}(\lambda_2) C_1^1 + 2C_2^1{}^\top \mathrm{diag}(\lambda_2) C_2^1 \\
H_3^C(\lambda_3) &= C_1^2{}^\top \mathrm{diag}(\lambda_3) C_1^1 + C_2^2{}^\top \mathrm{diag}(\lambda_3) C_2^1 + C_1^1{}^\top \mathrm{diag}(\lambda_3) C_1^2 + C_2^1{}^\top \mathrm{diag}(\lambda_3) C_2^2 \\
H_4^C(\lambda_4) &= 2C_1^2{}^\top \mathrm{diag}(\lambda_4) C_1^2 + 2C_2^2{}^\top \mathrm{diag}(\lambda_4) C_2^2 \\
H_5^C(\lambda_5) &= C_2^2{}^\top \mathrm{diag}(\lambda_5) C_1^1 - C_1^2{}^\top \mathrm{diag}(\lambda_5) C_2^1 + C_1^1{}^\top \mathrm{diag}(\lambda_5) C_2^2 - C_2^1{}^\top \mathrm{diag}(\lambda_5) C_1^2
\end{aligned}$$

The constraint Hessian (3.7) does not depend on variable u and only involves sparse matrix multiplication.

3.1.3 Discretising $\mathcal{S}(u)$

The continuous regularisers can be approximated by using suitable linear differential operators to build a discretised block matrix B for every instance of its continuous definition $\mathcal{B}$:

$$B^{Diff} = \begin{bmatrix} \partial_1^1 & 0 \\ \partial_2^1 & 0 \\ 0 & \partial_1^2 \\ 0 & \partial_2^2 \end{bmatrix} \quad B^{Curv} = \begin{bmatrix} \partial_{1,1}^1 + \partial_{2,2}^1 & 0 \\ 0 & \partial_{1,1}^2 + \partial_{2,2}^2 \end{bmatrix} \quad B^{Elas} = \begin{bmatrix} \sqrt{\mu}\,\partial_1^1 & 0 \\ \sqrt{\mu}\,\partial_2^1 & 0 \\ 0 & \sqrt{\mu}\,\partial_1^2 \\ 0 & \sqrt{\mu}\,\partial_2^2 \\ \sqrt{\lambda+\mu}\,\partial_1^1 & \sqrt{\lambda+\mu}\,\partial_2^2 \end{bmatrix}$$

Thus, the discretised regulariser function is given as follows:

$$S(u) = \frac{\alpha}{2}\|B\mathrm{d}u\|_2^2 \tag{3.8}$$

and its derivatives are:

$$\nabla S(u) = \alpha B^\top B \mathrm{d}u \qquad \nabla^2 S = \alpha B^\top B \tag{3.9}$$

3.1.4 Discretising $\mathcal{D}(u)$

Distance function $\mathcal{D}(T(u), R)$ uses the image interpolation function $T(u)$ to evaluate pixel intensities at the locations of pixels. Linear or spline interpolations can be used [69] where the degree of nonlinearity balances the quality against computational cost. As the pixel location is decided by its cell-centre, the staggered grid $u = (u^1(\blacktriangleright), u^2(\blacktriangle))$ is shifted to cell-centred grid $u(\bullet)$ for image interpolation function $T(u(\bullet))$:

$$D(u) = \frac{1}{2}\|T(Pu) - R\|_2^2 \tag{3.10}$$

and the derivatives are given by:

$$\nabla D(u) = P^\top \nabla T(Pu)^\top (T(Pu) - R) \qquad \nabla^2 D = P^\top \nabla T(Pu)^\top \nabla T(Pu) P \tag{3.11}$$

where the image gradient $\nabla T(Pu)$ is computed by the finite differences technique in discretised image analysis [69]. The Hessian $\nabla^2 D$ is defined without the second order image gradient $\nabla^2 T$ in order to ensure that it is positive-definite.

3.2 Optimisation

Let $F(u) = D(u) + S(u)$, the discretised constrained registration (3.1) can be written as a nonlinear constrained optimisation problem:

$$\begin{aligned} &\underset{u}{\text{minimise}} && F(u) \\ &\text{subject to} && Au = 0 \quad , \quad C(u) = 0 \end{aligned} \tag{3.12}$$

One popular approach to solve this nonlinear programming system is the SQP algorithm [81]. The SQP algorithm generates an iterative sequence:

$$\begin{bmatrix} u_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \lambda_k \end{bmatrix} + \tau_k \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} \tag{3.13}$$

where p_k is a direction of search and λ_k is the Lagrangian multipliers. p_k and p_λ are the solutions of the Newton-KKT system:

$$\begin{bmatrix} H_k & \begin{bmatrix} A \\ \nabla C(u_k) \end{bmatrix}^\top \\ A & 0 \\ \nabla C(u_k) & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla F(u_k) + \begin{bmatrix} A \\ \nabla C(u_k) \end{bmatrix}^\top \lambda_k \\ -Au_k \\ -C(u_k) \end{bmatrix} \tag{3.14}$$

where H_k is the approximated Hessian of the Lagrangian: $H_k = \nabla^2 F(u) + \nabla^2 C(\lambda)$. The Newton-KKT system (3.14) needs an inversion of matrix size $mn \times mn$. The updates generated should be feasible in the descent direction. One can take advantage of this observation to reduce the problem dimension. If u_k is feasible, then one of the conditions is $Au_k = 0$ and the search direction should satisfy:

$$Ap_k = 0 \tag{3.15}$$

Any vector p_k that satisfies (3.15) must be a linear combination of the columns of $Z \in \Re^{n\times(n-t)}$, a *null-space* of A. Therefore, the following equivalence holds:

$$Ap_k = 0 \iff p_k = Zp^z \tag{3.16}$$

for some $p^z \in \Re^{n-t}$. Since $A \in \Re^{t\times n}, t < n$, has full row rank, one can compute Z efficiently by QR decomposition [81]: $A^\top = QR$. Q is an orthogonal matrix and can be partitioned to $Q = [Y \quad Z]$, where $Y \in \Re^{n\times t}$ and $Z \in \Re^{n\times(n-t)}$ denote the range space and null space of A respectively. If an initial feasible point which lies in the null space of A is chosen, then it follows that:

$$\begin{aligned} Au_{k+1} &= Au_k + \tau_k Ap_k = 0 \\ AZu^z_{k+1} &= AZu^z_k + \tau_k AZp^z_k = 0 \qquad \forall p^z \in \Re^{n-t} \end{aligned}$$

The constrained optimisation problem (3.12) thus only needs to solve for a reduced variable $u^z \in \Re^{n-t}$:

$$\begin{aligned} \underset{u^z}{\text{minimise}} \quad & F(Zu^z) \\ \text{subject to} \quad & C(Zu^z) = 0 \end{aligned} \tag{3.17}$$

The iterative update (3.13) becomes:

$$\begin{bmatrix} u^z_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} u^z_k \\ \lambda_k \end{bmatrix} + \tau_k \begin{bmatrix} p^z_k \\ p_\lambda \end{bmatrix} \tag{3.18}$$

which requires solving the reduced Newton-KKT system for p^z_k and p_λ:

$$\begin{bmatrix} Z^\top H_k Z & Z^\top \nabla C(Zu^z_k)^\top \\ \nabla C(Zu^z_k)Z & 0 \end{bmatrix} \begin{bmatrix} p^z_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -Z^\top \nabla F(Zu^z_k) + Z^\top \nabla C(Zu^z_k)^\top \lambda_k \\ -C(Zu^z_k) \end{bmatrix} \tag{3.19}$$

The solution (p^z_k, p_λ) is unique if the Newton system (3.19) is well defined. In other words, if the following assumption is satisfied:

Assumption 3.2.1 *Assume the problem has the following properties:*

- *The reduced constraint Jacobian $\nabla C(Zu^z)Z$ has full row rank;*
- *The reduced Hessian $Z^\top HZ$ is positive definite on the tangent space of the constraints: $p^{z\ \top} Z^\top HZp^z > 0\ \forall p^z \neq 0, \nabla C(Zu^z)p^z = 0$.*

The Newton-KKT system enables the derivation of a practical framework based on a quadratic subproblem:

$$\begin{aligned} \underset{p^z}{\text{minimise}} \quad & \nabla F(Zu^z_k)^\top Zp^z + \tfrac{1}{2} p^{z\ \top} Z^\top H_k Z p^z \\ \text{subject to} \quad & \nabla C(Zu^z_k)Zp^z + C(Zu^z_k) = 0 \end{aligned} \tag{3.20}$$

If the assumption 3.2.1 holds then the solutions of problem (3.20) can be uniquely computed by the modified Newton system:

$$\begin{bmatrix} Z^\top H_k Z & Z^\top \nabla C(Zu^z_k)^\top \\ \nabla C(Zu^z_k)Z & 0 \end{bmatrix} \begin{bmatrix} p^z_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -Z^\top \nabla F(Zu^z_k) \\ -C(Zu^z_k) \end{bmatrix} \tag{3.21}$$

Solving the system (3.21) for large scales is an active research area. Here, the state of the art MINRES algorithm [82] with a preconditioner is used:

$$\begin{bmatrix} Z^\top H_k Z & Z^\top \nabla C(Zu^z_k)^\top \\ 0 & -\nabla C(Zu^z_k) Z (Z^\top H_k Z)^{-1} Z^\top \nabla C(Zu^z_k)^\top \end{bmatrix}$$

MINRES is an iterative Krylov subspace method that generates a solution of a linear system, $Lx = y$, with minumum Euclidean residual $\|Lx^* - y\|$. The reduced update p_k^z obtained from (3.21) should form a descent direction for the full update p_k and the stepsize τ_k should be accepted if sufficient decrease in a *merit* function is satisfied. In this case, a popular l_1 merit function is chosen:

$$\phi(u, \mu) = F(u) + \mu |C(u)| \tag{3.22}$$

A line search is then used to find a τ_k which sufficiently satisfies the decrease condition:

$$\phi(u_k + \tau_k Z p_k^z, \mu_k) \leq \phi(u_k, \mu_k) + \eta \tau_k \Delta(\phi(u_k, \mu_k); p_k^z), \qquad \eta \in (0, 1) \tag{3.23}$$

where $\Delta(\phi(u_k, \mu_k); p_k^z)$ denotes a directional derivative of ϕ in the direction p_k^z. It has been shown in [81, Theorem 18.2] that this search direction is a descent direction if the assumption 3.2.1 holds and the merit penalty parameter is sufficiently large. The method terminates when the change in u is less than a threshold ϵ_u. Algorithm 3.1 describes the method in detail.

```
Algorithm 3.1: SQP Constrained Registration $u \leftarrow \texttt{SQP}(D(T,R), S, C, \alpha, K, u_0)$
Set $u \leftarrow u_0,\ \lambda \leftarrow 0$;
Compute $Z$ as in (3.16);
for $k = 0, ..., K$ do
    Compute $D(T(u), R), S(u), C(u), \nabla F(u), \nabla C(u), H$;
    Set $u_{old} \leftarrow u, \quad \lambda_{old} \leftarrow \lambda, \quad Zu^z \leftarrow u_{old}$;
    Solve the system (3.21) for $p^z, \lambda$;
    Set $\mu \leftarrow \|\lambda\|_\infty + \epsilon$;
    $\tau \leftarrow$ LS($u_{old}, Zp^z, \nabla F(u_{old})^\top Zp^z - \mu|C(u_{old})|$,(3.23));
    Set $u \leftarrow u_{old} + \tau Z p^z, \quad \lambda \leftarrow \lambda_{\text{old}} + \tau(\lambda - \lambda_{\text{old}})$;
    if $\|u - u_{old}\| < \epsilon_u$ then
        break;
Return $u$;
```

```
Function LS($u, \delta u, \sigma$, condition)
$\tau \leftarrow 0.25, \quad \eta \leftarrow 10^{-5}$;
for $i = 0, ...,$ MaxIter do
    Set $u \leftarrow u + \tau^i \delta u$;
    if condition = true then
        break;
Return $\tau^i$;
```

3.3 Multilevel registration

In order to speed up the registration problem, Algorithm 3.1 is performed on a hierarchy of image data discretisation $h \in \{2^0, 2^1, ..., 2^H\}$. Starting with the coarsest level, $l = H$ or $h = 2^H$, we perform SQP to obtain a coarsest grid displacement u^H and compute a prolongation $u^{l-1} = \text{Prolong}(u^l)$, where this prolongation serves as the initial point for level $l - 1$. We repeat this process until we reach the finest level, $h = 2^0$. The advantages of this scheme are two folds:

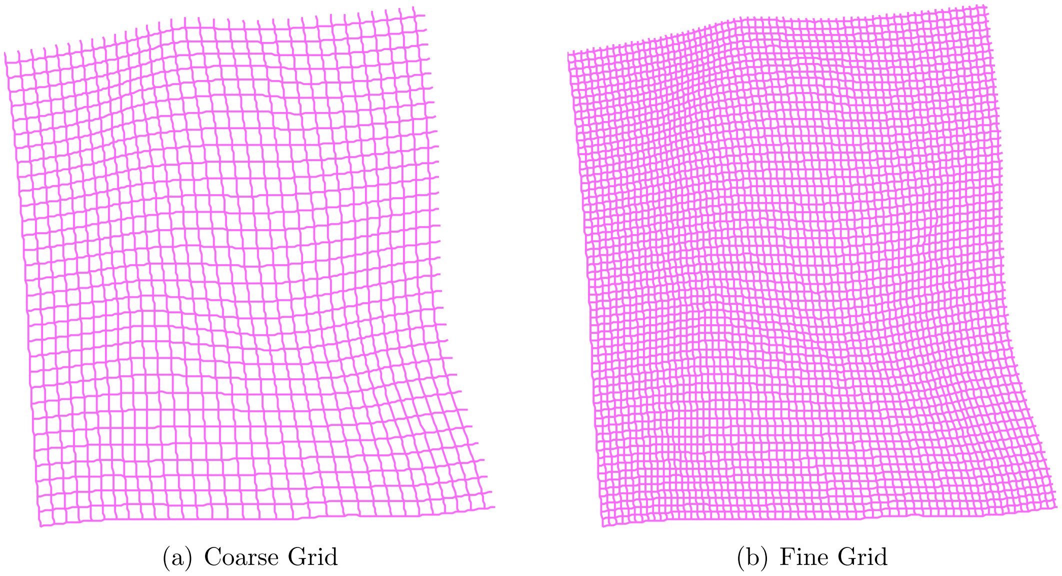
- Image registration is a highly nonlinear problem, sometimes ill-conditioned. Solving such a problem at a large scale is computational expensive and may fall into local minima. By using coarser levels, not only is the size of the problem reduced, but the nonlinearity of the problem is also relaxed. Multilevel representations of image data can be regarded as a different approach to smooth the image. Suppose that the measurement of an image pixel is the average light intensity on the corresponding cell plus some noise. The coarser image pixel is a smoothed measurement obtained by averaging adjacent cells. The result of this is to represent the image with fewer pixels, and thus solve a registration problem with fewer variables.

- A coarse registration often provides a good starting guess which is close the the local minimum of the finer level. Therefore, the convergence is accelerated by using Newton-type methods.

As the coarse registration smooths the problem, the regulariser parameter α increases as image data gets finer. The scheme is described in Algorithm 3.2 and the prolongation process is illustrated in Figure 3.3.

Algorithm 3.2: Multilevel Registration $u \leftarrow \texttt{MLV}(H, \alpha, K)$

```
Set $l \leftarrow H, u_0^l \leftarrow 0, \quad \kappa > 1$;
while $l \geq 0$ do
    Compute images $T, R$ with resolution $h = 2^l$;
    $u^l \leftarrow \texttt{SQP}(D(T, R), S, C, \alpha, K, u_0^l)$;
    if $l > 0$ then
        $u_0^{l-1} \leftarrow \texttt{Prolong}(u^l, l - 1)$;
        $l \leftarrow l - 1, \quad \alpha \leftarrow \kappa\alpha$;
```


(a) Coarse Grid

(b) Fine Grid

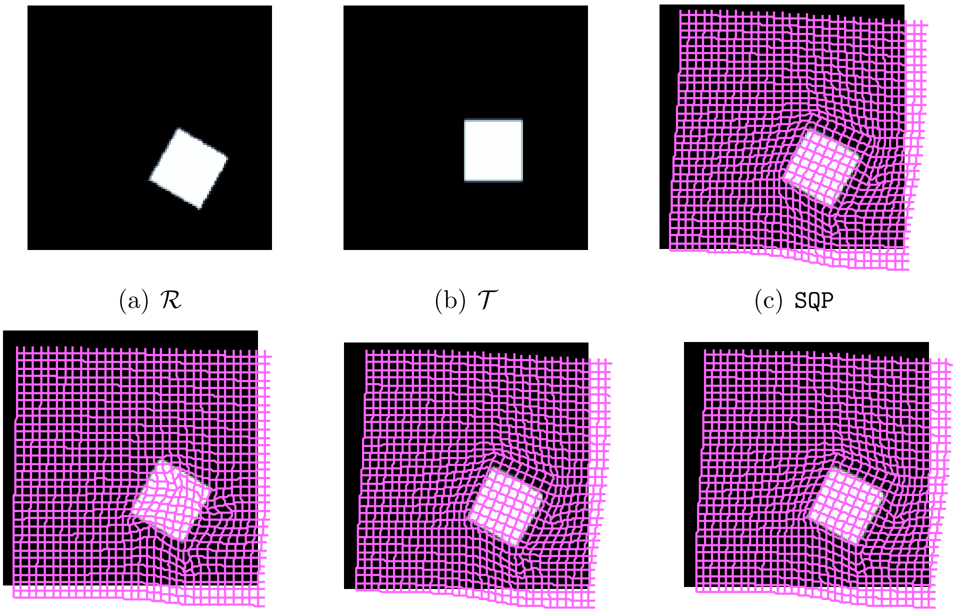
Figure 3.3: *Grid Prolongation:* `FineGrid` $\xleftarrow{Prolong}$ `CoarseGrid`

3.4 Experimental results

In this section, the proposed `SQP` is compared against the `Penalty` method of Modersitzki [70], where the constrained registration is described as an unconstrained optimisation problem:

$$(\texttt{Penalty}): \quad \underset{u}{\text{minimise}} \quad D\left(T(u), R\right) + S(u) + \rho \|W(Pu) \odot C(u)\|^2$$

In [70], the constraints are applied everywhere in the image subject to a predefined weighting function $W(Pu)$, i.e. at a rigid pixel, the weight is set to a higher value than at nonrigid pixel. One disadvantage of this method is the importance of a good estimate for the penalty parameter ρ. A large value of ρ guarantees rigid constraints, however it may result in less similarity between images. On the other hand, a small value of ρ encourages similarity of images but may fail to preserve the constraint. In contrast, in the method proposed, feasibility is enforced as hard constraints. As a result, solving the Newton system guarantees the feasibility at every iteration. This observation is benefitial for clinical applications. For instance, the movement of a bone should always be rigid. The proposed method is competitive in term of performance as we solve a reduced dimensionality problem.


(a) $\mathcal{R}$ (b) $\mathcal{T}$ (c) `SQP`

(d) `Penalty` : $\rho = 0.1$ (e) `Penalty` : $\rho = 10$ (f) `Penalty` : $\rho = 100$

Figure 3.4: *One Square: Transformations of* `SQP` *and* `Penalty` *with various parameters* ρ

In order to set up some benchmarks for comparison, define a ratio of reduction:

$$|D^*/D^0| = \frac{D(T(u^*), R)}{D(T, R)}$$

and constraint qualifications for `SQP` and `Penalty`:

$$C = |C(u)| \quad \text{and} \quad C = |W(Pu) \odot C(u)|$$

3.4.1 One square

Figure 3.4 illustrates the transformations produced by `SQP` and `Penalty` with various values for the penalty parameter ρ. It can be seen that for a small penalty weight, the constrained qualification C is too large. However, with larger ρ, the `Penalty` method favours the constraint and results in a reduction in registration accuracy (higher ratio $|D^*/D^0|$).

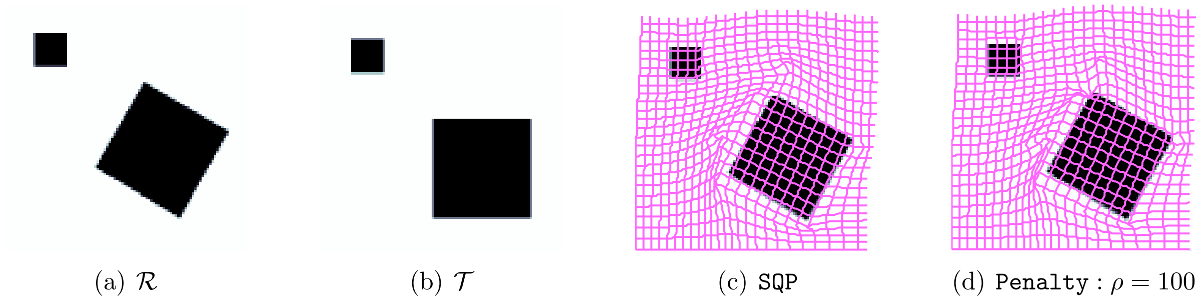


Figure 3.5: *Two Squares: Transformations of* `SQP` *and* `Penalty` *with various parameters* ρ

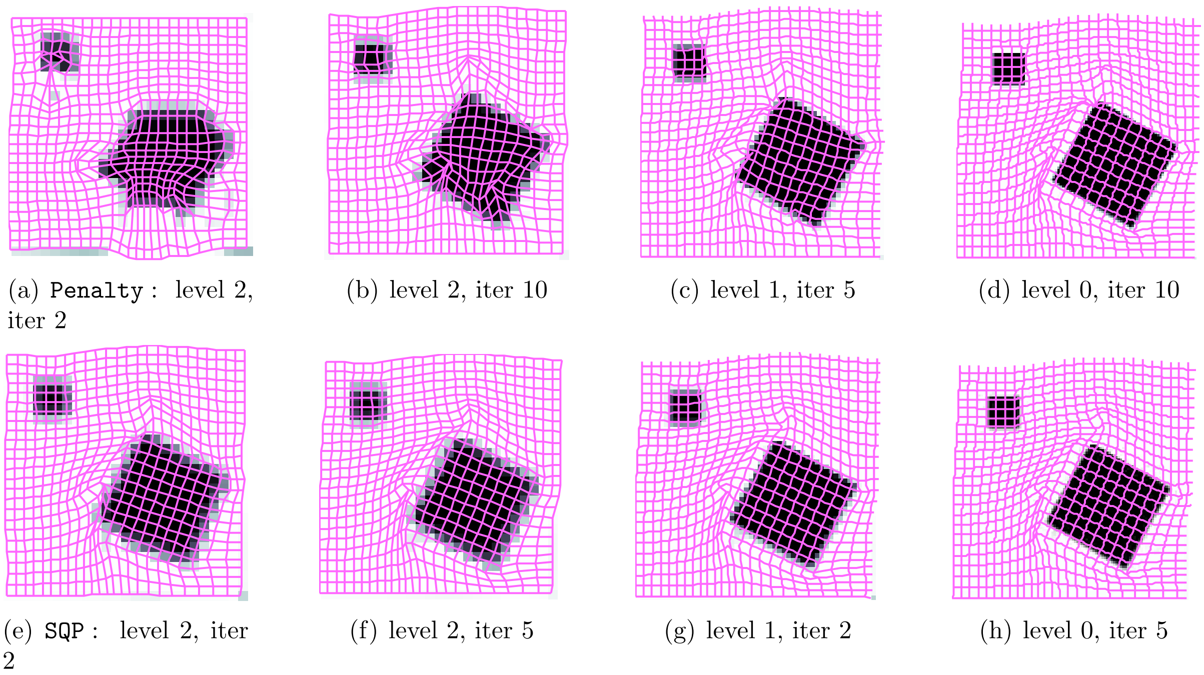


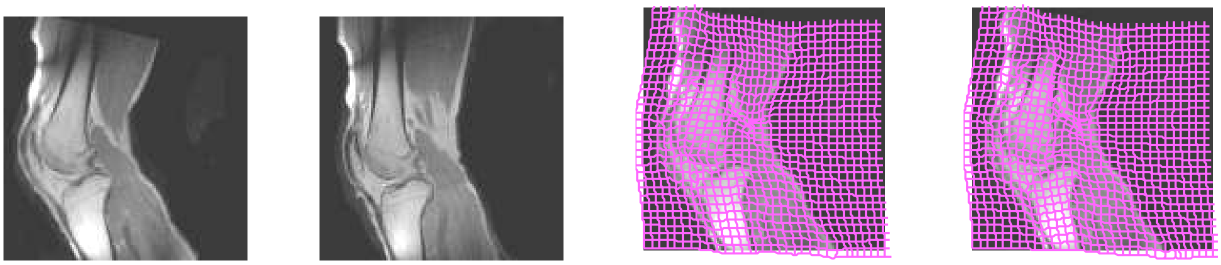
Figure 3.6: *Two Squares: Image grid transforming during the registration by* `SQP` *and* `Penalty`

3.4.2 Two squares

This experiment also shows similar patterns as the previous example, as shown in Figure 3.5 and Table 3.1. Figure 3.6 shows how the transformation grid looks like during the iterations at different levels. It demonstrates that the proposed method preserves feasibility at every iterations. Additionally, at a finer level, `SQP` produces smooth transformations within a few iterations. This is because very good results are obtained in the coarser levels.

Problem	Method	$\lvert D^*/D^0\rvert$	C	Iterations	Time (s)
One Square	`SQP`	0.47%	0.0006	24	9.3
	`Penalty` : $\rho = 0.1$	2.03%	239	21	8.7
	`Penalty` : $\rho = 10$	2.67%	9.4	26	10.2
	`Penalty` : $\rho = 100$	5.61%	0.4	28	10.5
Two Squares	`SQP`	1.12%	2.1	30	20.2
	`Penalty` : $\rho = 1$	1.25%	3000	25	18.4
	`Penalty` : $\rho = 10$	3.89%	261	40	37.2
	`Penalty` : $\rho = 100$	5.79%	4.5	32	24.6

Table 3.1: *Performance of* `SQP` *and* `Penalty` *with various* ρ



(a) $\mathcal{R}$ (b) $\mathcal{T}$ (c) `SQP` (d) `Penalty` : $\rho = 100$

Figure 3.7: *MR Knee: Transformations of knee joint motion by* `SQP` *and* `Penalty`
`SQP` : $\lvert D^*/D^0\rvert = 16.93\%$ $C = 2.9$
`Penalty` : $\rho = 100$ $\lvert D^*/D^0\rvert = 17.12\%$ $C = 4.5$

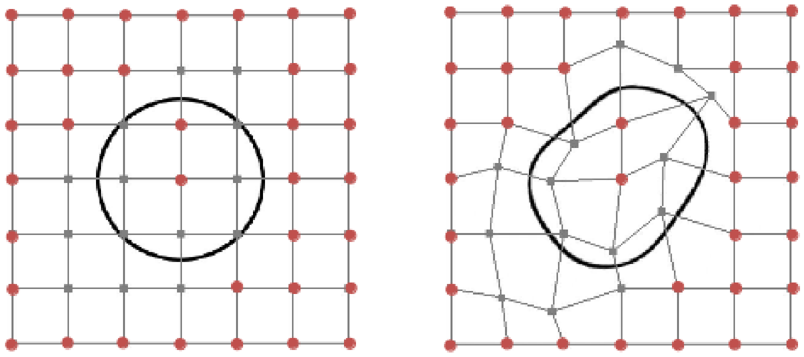
3.4.3 Knee joint

In this example, the effectiveness of the proposed `SQP` algorithm is demonstrated on a real image slice of the knee. The reference image $\mathcal{R}$ is in a bent position and the template image $\mathcal{T}$ is in an almost straight position. The knee has soft tissues that are free to deform and bones which must move rigidly. `SQP` produces a reduction ratio of 16.93%, a good figure compared to 17.12% of the `Penalty` method [70]. In addition, it does not require the estimation of the penalty parameter, maintain a low feasibility qualification, and performs within a competitive time compared with the `Penalty` method.

Chapter 4

Markov Random Fields (MRF) in image processing

In Chapters 2 and 3, we consider the image registration problem and its discretised model using the staggered grid discretisation method. A constrained optimisation problem is formulated and solved by the sequential quadratic programming (SQP) algorithm to obtain the optimal image transformation. The approach benefits from explicit controls of pixel movement and has good performance for problems with local rigid constraints. However, the SQP method becomes computationally impractical for high resolution input images because a fine discretisation for high resolution images results in extremely large scale constrained problem. As a result, the computation time of SQP may be too long and beyond the acceptable time for a medical application. Rueckert et. al. [90] have proposed a parametric framework that uses a few control points to coordinate pixel displacements. Figure 4.1 shows an example of an image transformed by moving the control points. The framework leads to a highly nonlinear problem that can be solved by continuous optimisation methods [52]. Although the approach overcomes the dimensionality problem, it still has some drawbacks, including convergence to local optima and expensive computational cost for certain justifiable choices for the objective function. Recently, Glocker et. al. [31] have employed discrete optimisation for solving the image registration problem. This is a discrete registration model based on on the parametric transformation [90] and Markov Random Fields (MRF). A discrete domain is defined with discrete displacements of the control points. Each control point displacement corresponds to a corresponding energy value. Each control point must admit one unique displacement. The objective



(a) Original Grid.

(b) Moving Grid.

Figure 4.1: *Pixel displacement by control points.*

function aims to minimise the total energy of all control points. This allows the interpretation of the registration problem as a discrete labelling problem. Interestingly, the discrete labelling problem is equivalent to an instance of MRF which is a powerful model in image processing. In this chaper, we introduce the probabilistic framework of random fields which is the basis of many mathematical models of image processing. The background discussion in this chapter is based on [11, 32, 56, 60, 65]. At first, we show that many problems in computer vision (such as image restoration, image segmentation, image stereo and image registration) can be seen as discrete labelling problems. It turns out that these discrete labelling problems belong to the discrete Markov Random Field (MRF) model. MRF has been used extensively in modelling for computer vision, artificial intelligence and machine learning. We provide a background to MRF and the Bayesian justification underlying the MRF model. We present the *Maximum a posteriori* principle, which is the objective in developing MRF minimisation methods. In addition, state-of-the-art algorithms for MRF minimisation are briefly reviewed focusing on its linear programming (LP) relaxation. The contribution of this chapter commences from the development of the dual decomposition technique for solving the LP relaxation. This leads to the development of proposed algorithms in Chapter 5.

4.1 Discrete labelling and Markov Random Fields

Many common image processing tasks require the inference of some hidden quantities x based on some observations d from the visual input data. One way to achieve the hidden quantities is to define a measure of goodness based on possible combinations of hidden quantities x and input data d:

$$F : (x, d) \to F(x, d)$$

This measurement represents how well a solution x fits into a given image processing problem with visual input data d. For any given instance of the image processing problem, the observations d remain fixed. Therefore, the problem is fully supported from an optimisation point of view, which chooses an optimal $\bar{x}$ that satisfies:

$$\bar{x} = \arg\min_{x} F(x)$$

where d is unchanged and implicitly attached in the function $F(x)$. The extensive use of the optimisation paradigm in vision is favoured by the fact that image data are often incomplete or some hidden information needs to be discovered. For instance, the sources of uncertainties can be image noise (due to imperfect sensors or quantisation errors), occlusions in the observed image or ambiguities in the visual interpretation. Based on this fact, perfect or exact solutions rarely exist. Instead, the true information can be approximated by inexact solutions which optimally satisfy the goodness of the measurement $F(x)$. In fact, it is due to the existence of these uncertainties that principles from statistics or probability theory are often used as the basis for deriving the exact form of the goodness measurement $F(.)$. Furthermore, the optimisation framework provides flexible spaces for additional quantisation measure and constraints to satisfy the nature of images.

4.1.1 Discrete labelling

Two issues have been raised in the optimised-based technique for image processing:

- The modelling of the hidden quantities x: how do we represent it, what constraints we need to impose on it, what is the expected accuracy of the model.
- The objective function $F(x)$: what is the goodness of fit we want, how complex the method can handle.

These two issues are interrelated and can have a great impact on how effective the optimisation process will be.

In this thesis, we represent the hidden quantities by a discrete set of labels, and the objective function by a graphical model. In this case, the problem is reduced to a discrete graph labelling problem. More specifically, whenever we refer to the term MRF we will hereafter mean a problem which is defined in terms of two basic entities: a graph G and a set of labels L.

The graph $G = (V, E)$ consists of a discrete set of nodes V[1] and a set of edges E . The nodes in V can represent:

- image features, e.g. a corner point or a line segment, on which a quantity must be estimated,
- image pixels, e.g. a discretised cell in an image that needs to be assigned an intensity value,
- image objects, e.g. predefined foreground, background, edges

while the edges E of the graph are used for encoding all relationships between the nodes of G. From a computational aspect, we process images digitally, i.e. finite number of features, objects, pixels in an image. Therefore, a graph with finite number of nodes and edges is sufficient to model the problem.

In addition, digital images contain finite quantised ranges of intensity. For instance, a simple gray-scale image allows intensity values within a discrete range of $[0, 255]$. These values can be considered as labels of a finite label set L, and are to be assigned for every node of the graph. The labels correspond to the hidden quantities that we want to estimate, e.g. intensities, disparities, foreground/background, or any other quantity of interest. Under these settings, the image procesing problem is reduced to a label assignment problem, that assign a unique label from the label set L to every node in V. In other words, we need to define a mapping x with domain V and range L, i.e. for every node $a \in V$:

$$x_a \overset{\text{def}}{=} x(a) : V \to L$$

The next issue is to formalise the objective function, where the chosen $F(x)$ should be able to encode all contextual constraints between the graph nodes. It turns out that a very good (and

[1] we use the terms nodes, objects, vertex interchangeably to regards an element of the set V

common) way for modeling these contextual constraints is by using discrete Markov Random Fields (MRF) [65, 114]. MRFs are built from the context of probability and belong to a particular type of probabilistic graphical models. More specifically, they form a class of undirected graphical models. A Markov Random Field model consists of two entities: a graph G and a set of labels L; and an objective function $F(x)$ that encodes the contextual information about the images. We hereafter use MRF model and discrete graph labelling interchangeably. A more rigorous background on MRFs is given in the next section along with justifications of the forms of their associated objective functions. Let us consider an objective $F(x)$ defined as follow:

$$F(x) = F_{unary}(x) + F_{pairwise}(x)$$

The first term is known as unary term and is defined as a cost to assign a label $l \in L$ to a node $a \in V$:

$$F_{unary}(x) = \sum_{a \in V} \theta_a(x_a)$$

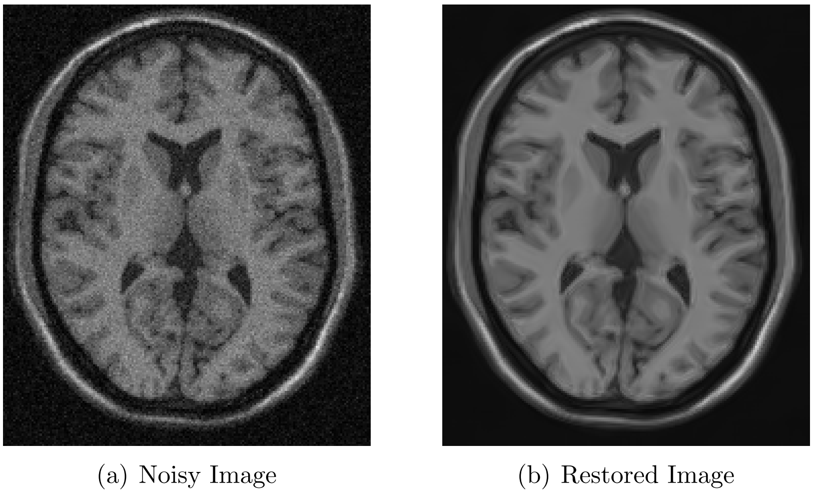
The unary term encodes how much the assignment of label x_a to node a disagrees with the observed image data at that node. On the other hand, the pairwise term is used to describe the contextual constraints between neighbouring nodes (the edges E) in the graph:

$$F_{pairwise}(x) = \sum_{ab \in E} \theta_{ab}(x_a, x_b)$$

The pairwise terms $\theta_{ab}(x_a, x_b)$ express our a priori knowledge about the hidden quantities of the nodes independently of the observation data. For instance, the assumption that neighbouring pixels should have similar intensities is a so called *prior*. Such priors impose constraints on the solution space of x. If no prior information is available, one assumes an uniform distribution, where every labeling has equal prior probability. The general form of the objective function is given by:

$$F(x) = \sum_{a \in V} \theta_a(x_a) + \sum_{ab \in E} \theta_{ab}(x_a, x_b) \tag{4.1}$$

Despite this seemingly simple formulation of the objective function associated to an MRF. MRFs are capable of capturing numerous problems in computer vision, machine learning and artificial intelligence. Such problems are discussed in the following illustrative examples:



(a) Noisy Image (b) Restored Image

Figure 4.2: *Image restoration: we are given an image corrupted with noise and try to recover a smooth image that should be as similar as possible to the true original image*

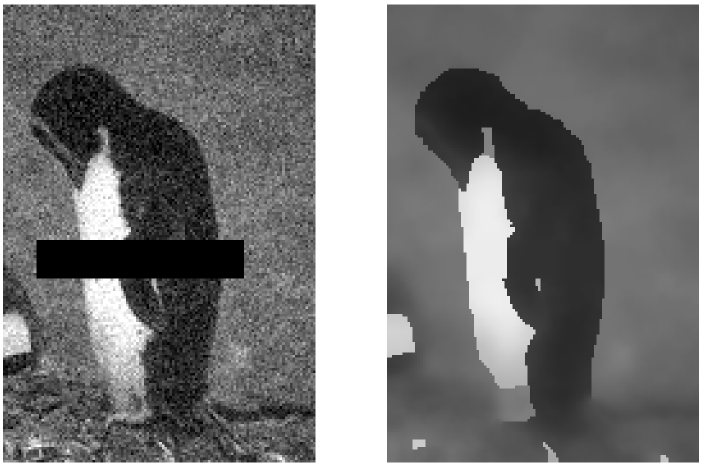
Image restoration and inpainting [50, 68, 102]

Assume we have an input digital image with corrupted parts and noises (e.g. due to bad lighting conditions at the time of capturing). Before trying to infer any higher level information from that image, it would be necessary to remove the noise and restore the original content of the image, as shown in Figure 4.2 and Figure 4.3. In other words, we seek to find the true underlying pixel intensities (or colours if we are dealing with colour images). The first step in MRF modelling is to define random variables x. Assuming the digital image I contains n pixels, then our graph has n nodes, i.e. $|V| = n$. The edges $ab \in E$ represent the neighbouring pixels in horizontal and vertical directions. Thus, the graph $G = (V, E)$ coincides with the image grid. The unknown quantities here are the true intensity of image pixels, corresponding to the label set (for example, for 8-bit gray-scale images, $L = \{0, 1, ..., 255\}$). The unary terms are defined so as to express the fact that the restored intensity x_a at any pixel $a \in V$ should be close to the observed intensity $I(a)$:

$$\theta_a(x_a = l) = |I(a) - l| \quad \forall l \in L$$

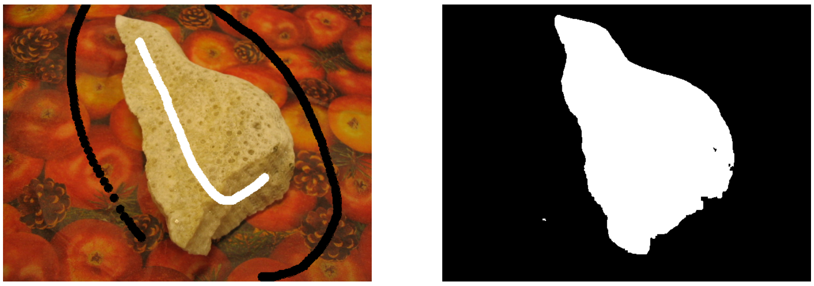
The pairwise potential function reflects the prior knowledge that the neighbouring pixels should have similar intensities. This seems valid everywhere in the image except for the boundary pixels between different objects. It is common to use the truncated semimetric:

$$\theta_{ab}(x_a = l, x_b = k) = \min(|l - k|, M) \quad \forall ab \in E, \forall l, k \in L$$



(a) Corrupted Image (b) Inpainted Image

Figure 4.3: *Image restoration and inpainting: we are given an image corrupted with noise and missing parts. We try to recover a smooth image and complete the missing parts.*



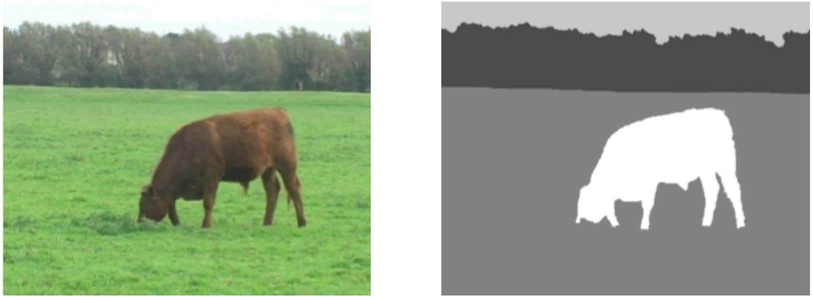
(a) Original Image (b) Binary Segmentation

Figure 4.4: *Binary Segmenation: In the original image, a user has marked a white line and a black line to distinguish between two regions.*

where M is the maximum value to penalise the neighbouring relationships.

Image segmentation [15, 1, 53, 102]

In this example, we have the input image I and a similar graph model as in the previous example, i.e. the graph G coincides with image grid. The aim of image segmentation is to identify specific objects or regions of an image by their distinct features. From a low-level perspective, this can be achieved by labelling individual image points to a set of predefined features L. This task can be naturally formulated as a multilabelling problem, where the set of labels is finite $L = l_1, l_2, ..., l_M$ and every pixel $a \in V$ is represented by one random variable x_a. As before, a common image grid can be used to describe the pairwise relationships between neighbouring pixels. The cost to assign a label $l \in L$ to a point $a \in V$ is encoded in the unary



(a) Original Image (b) Multilabel Segmentation

Figure 4.5: *Multilabel Segmenation: Image segmentation using unsupervised method with 4 labels: the field, the cow, trees and the sky.*

potential function. For example:

$$\theta_a(x_a = l) = -\log \rho(I(a)|a = l) \quad \forall a \in V,\ \forall l \in L$$

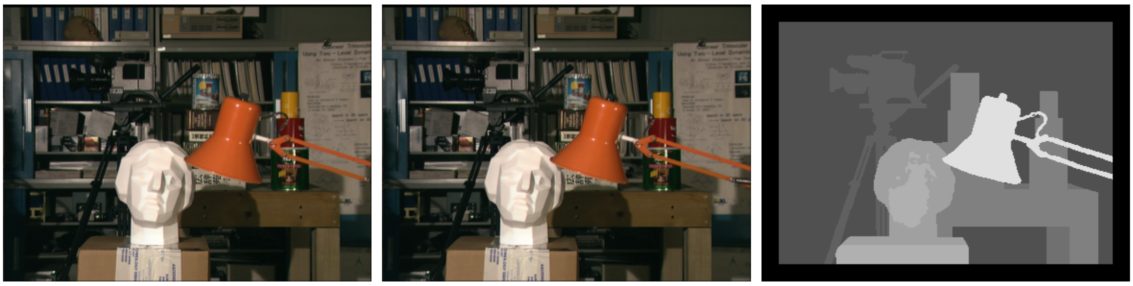
The unary potential makes use of predertermined probability distribution. If we want to assign a certain label x_a to an image pixel a, the unary terms evaluate how likely that label is, with respect to the intensity value $I(a)$. In addition, prior constraints can be encoded on the pairwise potential as:

$$\theta_{ab}(x_a = l, x_b = k) = \exp\left(-\frac{|I(a) - I(b)|}{\sigma^2}\right) \cdot \frac{1}{\|l - k\|} .(l \neq k) \quad \forall ab \in E,\ \forall l, k \in L$$

where $(l \neq k) = \{0, 1\}$ and σ corresponds to the level of noise in the image. The prior favours the same label for neighbouring pixels by assigning these with zero cost, i.e. if $x_a = x_b$, then $\theta_{ab}(x_a, x_b) = 0$. The cost for assigning different labels, corresponding to a boundary between pixels a and b, depends on the intensity difference of these two pixels. A simple binary segmentation can be computed exactly by a MRF model [15]. However, finding an exact solution for multilabelling is known to be NP-hard. Instead, partially optimal solutions can be computed by relaxation methods [53, 102]. Examples of binary segmentation and multilabel segmentation are shown in Figure 4.4 and Figure 4.5.

Stereo matching [101, 102, 107]

Another classical example of MRF usage is the image stereo matching problem. We are given a pair of two images, a left and a right image, captured by two digital cameras. Both images are located at the same height and look towards the same direction. We want to find for each pixel



(a) Left Image (b) Right Image (c) Disparity Map

Figure 4.6: *Stereo Matching application.*

of the left image its horizontal displacement, also known as disparity, in the right image (see Figure 4.6). Stereo systems reveal three-dimensional information about a scence. The depth or distance of an object to the observer is proportional to its disparity observable in the two views. Through identification of point correspondences in the images, we can determine these disparities and compute a dense depth map via triangulation. In order to formulate the stereo matching as a discrete labeling problem, we assume a finite number of depth layers. A set of labels can be defined as $L = \{l_1, ..., l_M\}$, i.e. a discrete set of potential disparities. Again, every pixel is a random variable and the graph coincides with the image grid. Unary terms measure the difference between corresponding pixels in the left and right image:

$$\theta_a(x_a = l) = |I_{left}(a + l) - I_{right}(a)|$$

If the disparity estimation is based only on the optimization of the data terms, then the result is likely to be noisy. In order to avoid this, we need to impose a smoothness contextual constraint via pairwise potential. Similar to the previous example, the same disparity for neighbouring pixels, known as the Potts model, is used:

$$\theta_{ab}(x_a = l, x_b = k) = (l \neq k) \quad \forall ab \in E, \ \forall l, k \in L$$

Image registration [31]

Image registration is one of the fundamental techniques in medical imaging whereby one tries to match a template image T to a reference image R. In the previous chapters, we discussed the dense image registration problem with constraints. However, for a large scale application, e.g. 3D image registration, computing dense displacements at every voxel (3D pixel) is prohibitively expensive. An elegant way of replacing dense displacements is by reparametersing them with a set of control points V. Clearly, the number of control points is much fewer than the number of image pixels, i.e. $|V| \ll |T|$. The dense displacement field can then be defined as a linear combination of control point displacements (Figure 4.1):

$$\delta(p) = \sum_{a \in \mathcal{N}(p)} \omega_a(p) x_a$$

where $\delta(p)$ represents dense displacement at image pixel $p \in I$, x_a denotes the displacement of control point $a \in V$, and $\omega_a(p)$ is some weighting functions. A popular concept in parametric image registration is based on free-form deformation (FFD) [90] using cubic B-splines weighting function ω_a. Here, ω_a determines the influence of a control point a to the image pixel p. In the FFD model, only certain local control points around the neighbourhood of image pixel p affect its displacement. A benefit of this parametric model is a significant reduction in problem size. We can compute a smooth dense displacement field for every pixel by manipulating a few control points. To this end, the hidden quantities in the MRF model are the displacements of control points, which are given by a set of discrete labels L. In this case, each label represents an acceptable displacements, for examples, $L = \{1\text{mm}, 2\text{mm}, ..., 10\text{mm}\}$ in a metric system. For every displacement $l \in L$ of a control point $a \in V$, we can compute the corresponding dense displacement for the image pixels that belong to the image patch $I(a)$, thus we obtain a transformed image patch $I(x_a = l)$. This leads to a definition of unary terms for the MRF model:

$$\theta_a(x_a = l) = \sum_{p \in I(a)} \phi(|a - p|).D(R, T(x_a = l))$$

where $D(.,.)$ evaluates the dissimilarity between R and T, of the pixels belong to the image patch $I(a)$ with the patch centered at the control point a. The function $\phi(.)$ denotes the influence of the control point a on the image pixel p. A simple pairwise term penalises the large displacement between neighbouring control points as given below:

$$\theta_{ab}(x_a = l, x_b = k) = \lambda_{a,b}|(x_a^{k-1} + l) - (x_b^{k-1} + k)|$$

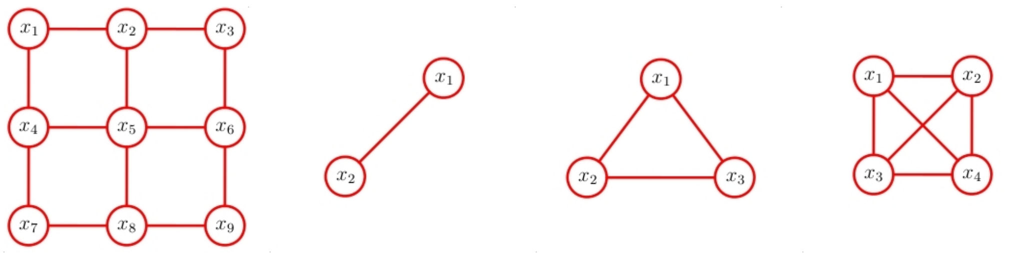
This pairwise potential works similarly to the elastic regulariser that has been mentioned in section 2.2. In [62], an approximated curvature regulariser, using the second order derivatives, is proposed with higher order MRF model.

4.1.2 Markov Random Fields (MRF)

It was shown earlier that many image processing problems can be modelled as a multilabelling problem. As will be shown, the solution of multilabelling problem is exactly the maximum a posteriori (MAP) estimation, an estimation that is based on likelihood criterion (unary term in the objective function of multilabel problem) and prior information (pairwise term in the multilabel problem). In order to estimate the MAP, one needs to find a model that describes the prior probability efficiently. This prior should encode all contextual constraints of the given problem. As we shall see, Markov Random Fields provide an efficient way to perform this task. A MRF model coincides with the structure of an image processing problem in such a manner that spatial interactions between objects are taken into account. The prior information is encoded in the local neighbourhood of objects. Therefore, it is unsurprising that the objective function of the MRF model contains basic probabilistic justifications for the objective of the multilabelling problem.

Markov Random Fields is a particular case of undirected graphical models. Consider a discrete set of labels L and an undirected graph $G = (V, E)$ consisting of $|V|$ nodes. A random field X forms a set of $|V|$ random variables x, where each variable $x_a \in X$ corresponds to a node $a \in V$ and can take a value from the label set L. Note that, in general each variable could have its own predefined set of labels L_a. However, in many applications, where the variables represent the same type of entity, they access a common label set L. Once every variable is assigned a label, this is knowon as a labeling of the field (sometimes a labelling is refered to a configuration or realisation of the field). A labelling, which can be seen as the occurence of a certain event, has a certain probability $p(x)$. This probability is often regarded as posterior distribution, which is dependent on the likelihood distribution (which is encoded into the unary potential) and the prior distribution (pairwise potential). A valid MRF distribution $p(x)$ should respect the probabilistic dependencies implied by the neighbourhood systems of the graph. The following describes the neighbourhood relationships of a graph before presenting the properties of a valid MRF model.

Neighbourhood systems Figure 4.7(a) gives a visual illustration of a first order random field model. The graph nodes V correspond to the random variable set X. The edges E encode the neighbourhood systems on a set of nodes. A clique is a subset of nodes $C \in V$, where every



(a) Regular grid (b) First Order (c) Second Order (d) Third Order

Figure 4.7: *Undirected Graphical Models: neighbourhood system and order.*

node is directly connected to all other nodes. So a clique is either a single node, or it constitutes a fully-connected subgraph. The order of a random fields is defined as the maximum clique size in a graph minus one. For instance, a regular grid imposes a maximum clique of size two and order one. It is the most common random fields model in image processing.

Markov Random Fields properties A valid MRF model should respect the following conditional independencies:

Definition 4.1.1 (Markov Random Field) *A random field x is said to be a Markov Random Field with respect to its neighbourhood system, if it satisfies:*

- *$p(x) > 0$, for all possible labelling.*
- *$p(x_a|x_{N(a)}) = p(x_a|x_{V_{-a}})$, where $N(a)$ represents all neighbouring nodes of $a \in V$ while V_{-a} denodes all nodes of V except a.*

The first property ensures that the joint probability can be uniquely determined by requiring that any labeling has a strictly positive probability. This property is usually satisfied in practice, or can be easily ensured. The second condition simply states the fact that any node in the graph G depends only on its immediate neighbours. The latter is exactly what allows Markov Random Fields to model contextual constraints between objects in an efficient manner, since all contextual constraints are now enforced only through local interactions between neighbouring nodes in G. This constitutes a very important property of MRF and is a key reason why MRFs have gained much popularity.

Maximum a posteriori (MAP) It was earlier mentioned that a labelling solution $\bar{x}$ turns out to be exactly the MAP estimation, which maximises the posterior distribution of x given all distributions of d. This section justifies the argument via Bayes's theorem. The rules of probability allow us to derive a connection between the conditional probability and the joint probability, i.e. $p(x, d) = p(x|d)p(d)$. Thus, the probability of a distribution x given observations d is given by $p(x|d) = p(x, d)/(y)$. Using the symmetry property of the joint distribution, Bayes Theorem gives:

$$p(x|d) = \frac{p(d|x)p(x)}{p(d)}$$

Since d is a constant, the term $p(d)$ can be dropped, therefore we have the relationship:

$$p(x|d) \propto p(d|x)p(x)$$

In the Bayes' terminology, $p(x|d)$ is the posterior distribution, $p(d|x)$ is called likelihood distribution and $p(x)$ is the prior distribution. According to MAP estimation, we choose to assign the labels to the random variables x, which maximise the posterior distribution $p(x|d)$ given all the observations d, i.e.:

$$\begin{aligned} \bar{x} &= \arg\max_x p(x|d) \\ \bar{x} &= \arg\max_x p(d|x)p(x) \qquad (4.2) \end{aligned}$$

So the MAP estimation is equivalent to the solution of the maximum probability of the product of likelihood distribution and prior distribution. The prior distribution $p(x)$ reflects a priori knowledge about the hidden variables independently of the observation d, and it is encoded in the MRF model. This knowledge is available before we obtain any observation. For instance, the assumption that neighbouring pixels should have similar intensities is a prior. Such priors impose constraints on the solution space of x. The likelihood distribution $p(d|x)$ evaluates how well a certain labeling of the hidden variables fits the observation. To this end, we can see that the multilabel problem is equivalent to the MAP estimation that can be modeled by Markov Random Fields. But what is the exact form of such distributions in the Markov Random Field model? This question is addressed by incorporating an additional concept in the following section.

Markov-Gibbs equivalence Spatial, contextual interactions on lattice graphs have a broad range of applications in various fields of statistical science. The origin of MRF framework can be dated back to physics. In the early 1920s, Ernst Ising, a German physicist and student of Wilhelm Lenz, developed a mathematical model for ferromagnetism in solid state bodies. Ising defined a set of nodes equally distributed on a rectangular domain; each node corresponds to a dipole which at any given moment is in one of two states, up or down. He derived the probabilities for the configurations of the field to be given by a Gibbs distribution.

Definition 4.1.2 (Gibbs random field) *A random field is said to be a Gibbs random field if and only if its joint distribution $p(x)$ is a Gibbs distribution, which has the following form:*

$$p(x) = \frac{1}{Z}.\exp\left(-\sum_{C\in G}\theta_C(x_C)\right) \tag{4.3}$$

where Z (also known as the partition function) is a normalising constant to ensure the sum of probabilities is equal to one:

$$Z = \sum_{C\in G}\exp\left(-\sum_{C\in G}\theta_C(x_C)\right)$$

In the equation (4.3), C denotes a clique in the graph G. The symbol $\theta_C(x_C)$ represents the clique potential, where each $\theta_C(x_C)$ is a real function that depends only on the random variables contained in clique C. The clique potential θ_C is not restricted to any specification, and the smaller the sum of all potentials, the higher the probability mass $p(x)$. Thus, it leads to the motivation to formulate θ_C as a cost of label assignment. Today, we know that the Gibbs random field is equivalent to the Markov Random Field, thanks to the proof of Hammersley and Clifford [40, 10].

Markov-Gibbs equivalence [40, 10] A distribution $p(x)$ over a discrete random field x is a Gibbs distribution (4.3), if and only if the random variables x make up a Markov Random Field with respect to the graph G.

The practical value of the Markov-Gibbs equivalence provides a tool to define the joint probability function of a Markov Random Field. It allows to define, determine, manipulate, and infer the underlying probability distributions in a convenient way. These functions encode all desired contextual constraints between labels and choosing the proper form for these functions constitutes the most important stage during MRF modeling. This thesis deals with the first order MRFs, therefore potential functions for all cliques with more than 2 elements will be assumed to be zero. Actually, the cliques of size 1 correspond to the nodes $a \in V$, and the cliques of size 2 correspond to the edges $ab \in E$ of the graph G. Therefore, the joint probability distribution (or Gibbs distribution) defined by the first order MRF model will have the following form:

$$p(x) = \frac{1}{Z}. \exp\left(-\sum_{a \in V} \theta_a(x_a) - \sum_{ab \in E} \theta_{ab}(x_a, x_b)\right) \tag{4.4}$$

Clearly, the sum over node potentials θ_a corresponds to the unary terms in multilabel problem or the likelihood distribution in the MAP estimation. The sum over edge potentials θ_{ab} corresponds to the pairwise terms or prior distributions. Dropping the constant Z and substituting (4.4) into (4.2) gives:

$$\begin{aligned}
\bar{x} &= \arg\max_x \quad \exp\left(-\sum_{a \in V} \theta_a(x_a) - \sum_{ab \in E} \theta_{ab}(x_a, x_b)\right) \\
\bar{x} &= \arg\max_x \quad \log\left[\exp\left(-\sum_{a \in V} \theta_a(x_a) - \sum_{ab \in E} \theta_{ab}(x_a, x_b)\right)\right] \\
\bar{x} &= \arg\min_x \quad \sum_{a \in V} \theta_a(x_a) + \sum_{ab \in E} \theta_{ab}(x_a, x_b) \qquad (4.5)
\end{aligned}$$

showing that the objective function of the MRF model (4.5) is exactly equivalent to the discrete multilabelling problem (4.1). Having defined the MRF models and the probabilistic justification underlying the image processing problem, the next section turns to the matter of optimising this model.

4.2 MRF optimisation techniques

The use of MRF has become increasingly popular in many kinds of imaging and vision applications. This can be attributed to its powerful characteristics in representing the image problem and more importantly, the recent advances in MRF optimisation algorithms that allow efficient computation for very large scale problems. Recent research has shown that some of the top methods for image stereo matching, segmentation, registration utilise MRF minimisers. Current state-of-the-art methods for MRF minimisation include graph-cut, belief propagation (BP) and message-passing/linear programming (LP) relaxation. While graph-cut and belief propagation are combinatorial-based optimisation, the message-passing algorithms rely on the theory of continuous optimisation (BP can be seen as a special case of message-passing techniques). Message-passing techniques are related to the dual of the linear programming (LP) relaxation but they are built in the sense of dynamic programming. As a result, they do not have complete convergent properties. This issue can be addressed via the dual decomposition technique. Dual decomposition is a common method in optimisation and it guarantees to converge to the global optinum of the primal problem.

In the following sections, we briefly describe graph-cut, BP, message-passing and LP relaxation. In particular, we focus on the dual decomposition of the LP relaxation, a technique that reformulates the original MRF model to a nondifferentiable optimisation problem.

4.2.1 Graph-cut

Graph-cut works similarly to greedy iterative algorithms. Two popular graph-cut algorithms, the swap-move algorithm and the expansion-move algorithm, were developed by Boykov et. al. [17]. Both algorithms iteratively select the best solution in the inner loop via a binary labeling problem which, in turn, is reduced into the problem of finding the minimum cut in an appropriately constructed capacitated graph. This process converges rapidly and results in a strong local minimum, in the sense that further moves will not produce a labeling with lower energy. For a pair of labels $\{\alpha, \beta\}$, a swap move takes some subset of the pixels currently given the label α and assigns them the label β and vice versa. The swap-move algorithm finds a local minimum such that there is no swap move, for any pair of labels $\{\alpha, \beta\}$ that will produce a lower energy labeling. Analogously, an expansion move for a label α increases the set of pixels that are given this label. The expansion-move algorithm finds a local minimum such that no expansion move, for any label α, yields a labeling with lower energy. The main computational cost of graph cuts lies in computing the minimum cut, which is done via the max-flow problem [16]. The max-flow problem defines a graph with two distinguised vertices, the source and the sink (in the binary graphcut problem, they are indeed the two labels). It then seeks to find the maximum amount of flow that can leave the source and arrive at the sink, while passing

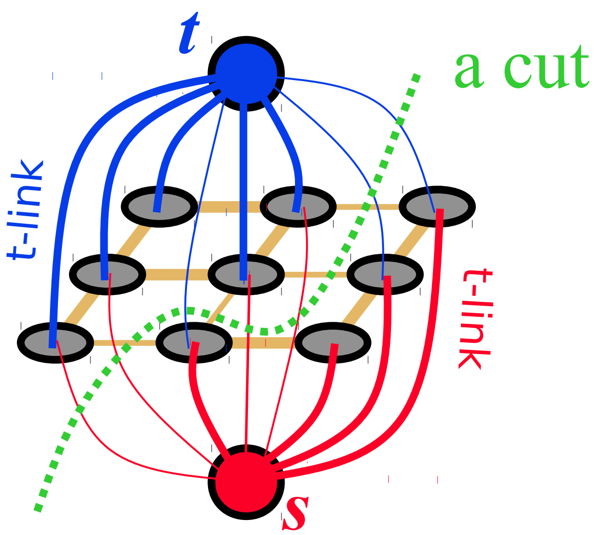


Figure 4.8: *A minimum cut that seperates nodes to either the source s or the sink t*

through any of the edges of the graph. In this case, the weight of an edge is interpreted as the edges capacity, i.e. it represents the maximum flow that can pass through that edge. It is well-known that the maximum amount of flow from the source to the sink equals the cost of the minimum cut that separates the source and the sink. This classification of nodes to either the source or the sink is equivalent to the binary labelling. Figure 4.8 illustrates the minimum cut problem.

Although graph-cut algorithms have shown superior performance in many computer vision applications, their use is limited to a restricted class of MRF. In particular, the expansion-move algorithm was shown to be applicable to any energy where θ_{ab} is a metric, i.e. it satisfies the following properties for any triplet of labels $\{\alpha, \beta, \gamma\}$:

$$\begin{aligned} \theta_{ab}(\alpha,\beta) &\leq \theta_{ab}(\alpha,\gamma) + \theta_{ab}(\gamma,\beta) \\ \theta_{ab}(\alpha,\beta) &= 0 \iff \alpha = \beta \\ \theta_{ab}(\alpha,\beta) &= \theta_{ab}(\beta,\alpha) \end{aligned}$$

while the swap-move is only applicable to semi-metric energies (metrics where the triangle inequality needs not hold). Kolmogorov and Zabih [55] subsequently relaxed these conditions and showed that the expansion-move algorithm can be used as long as:

$$\theta_{ab}(\alpha,\alpha) + \theta_{ab}(\beta,\gamma) \leq \theta_{ab}(\alpha,\gamma) + \theta_{ab}(\beta,\alpha)$$

and the swap-move algorithm can be used if for all labels $\{\alpha, \beta\}$, it satisfies:

$$\theta_{ab}(\alpha, \alpha) + \theta_{ab}(\beta, \beta) \leq \theta_{ab}(\alpha, \beta) + \theta_{ab}(\beta, \alpha)$$

If the energy does not obey these constraints, graph-cut algorithms can still be applied by truncating the violating terms [89]. In this case, however, it is no longer guaranteed to find the optimal labeling with respect to expansion or swap moves. In practice, this technique seems to work well only when relatively few terms need to be truncated [102].

4.2.2 Message passing

Belief Propagation (BP) is an iterative algorithm, which works by continuously propagating local messages between the nodes of the MRF model. There are two variants of BP: max-product BP and sum-product BP. While max-product BP is designed to find the lowest energy solution, the sum-product BP does not directly search for a minimum energy but instead computes the marginal probability distribution of each node in the graph. Both algorithms utilise the same message-passing concepts. We focus on the the max-product BP as its solution returns the labelling for the MRF problem. At every iteration, each node exchanges messages with all neighbouring nodes. This process is repeated until all messages are stabilised. Within the max-product class, there are different implementations available based on the schedules for passing messages on the grids. For instance, in [103], messages are passed along rows and then along columns. When a row or column is processed, the algorithm starts at the first node and passes messages in one direction (similar to the forward-backward algorithm for Hidden Markov Models). Once the algorithm reaches the end of a row or column, messages are passed backward along the same row or column.

So, what are the messages and how are they computed? The set of messages sent from a node a to a neighbouring node b will be denoted by $\{m_{ab}(x_b)\}_{x_b \in L}$. Therefore, the total number of such messages is always $|L|$ (i.e. there exists one message per label in L). Intuitively, the meaning of the message $m_{ab}(x_b)$ is that it expresses how likely node a thinks that node b should be assigned label x_b. Furthermore, the message $m_{ab}(x_b)$ is computed as:

$$m_{ab}(x_b) \leftarrow \min_{x_a \in L} \left(\theta_a(x_a) + \theta_{ab}(x_a, x_b) + \sum_{c \in \mathcal{N}(a)_{-b}} m_{ca}(x_a) \right)$$

In order to keep consistent notations, the above formulation takes a form of negative log updates for messages m_{ab} instead of the original max-product message updates:

$$\hat{m}_{ab}(x_b) \leftarrow \max_{x_a \in L} \left(\rho_a(x_a) \rho_{ab}(x_a, x_b) \prod_{c \in \mathcal{N}(a)_{-b}} \hat{m}_{ca}(x_a) \right)$$

where $\rho_a \propto \exp(-\theta_a)$ and $\rho_{ab} \propto \exp(-\theta_{ab})$ are based on likelihood and prior distribution respectively. The aim of the message is that node a reveals what it thinks about the labeling x_b. There are three factors to be considered in the process:

- Assuming node b is assigned x_b, node a has to consider what the best assignment x_a for itself. This is measured by the cost of pairwise function $\theta_{ab}(x_a, x_b)$.
- If label x_a is the most compatible one, node a has also to consider what is the likelihood of this label; this is measured by $\theta_a(x_a)$.
- Finally, node a needs to consider what its neighbours think about the label x_a. This is evaluated by considering all incoming messages from neighbouring nodes (except of b).

Once every node has sent and received a sufficient amount of messages, based on the beliefs we can compute the configuration of the field:

$$\bar{x}_a = \arg\min_{x_a \in L} \quad \theta_a(x_a) + \sum_{b \in \mathcal{N}(a)} m_{ab}(x_a)$$

The original BP proposed by Pearl [84] was intended to be used only for graphs without cycles, such as Bayesian networks. On acyclic graphs, BP guarantees to find the global optimum solution. Furthermore, it can be shown that this global optimum may be computed in just one iteration. However, there is nothing in the formulation of BP that prevents it from being used on graphs with loops. Indeed, BP has been successfully applied to cyclic graphs in quite different problem domains such as early vision [28] and error-correcting codes [29]. In general, Loopy-BP is not guaranteed to converge and may go into an infinite loop switching between two labelings. However, if Loopy-BP converges and there are no ties in the min-marginals for nodes, it has a strong local minimum property that is somewhat analogous to that of graph cuts [109, 111].

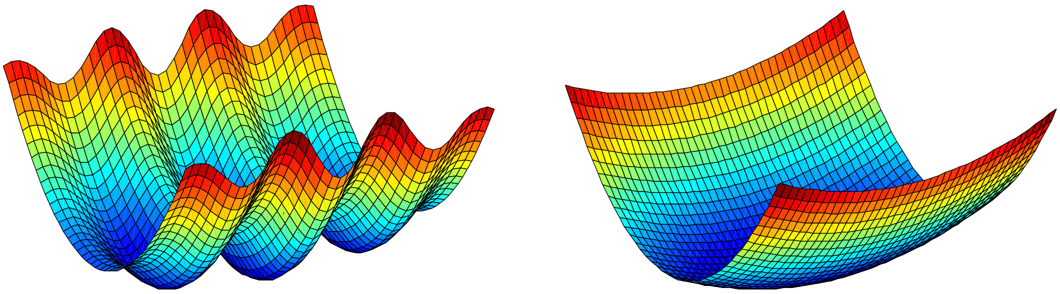
TreeReWeighted message passing (TRW) is similar to BP in term of message-passing. However, the message-passing mechanism is based on the iterative solutions of the underlying linear programming relaxation. The LP relaxation interpretation will be discussed in the next section. Here, we discuss the message-passing properties of TRW. In TRW, the message update rule is defined by:

$$m_{ab}(x_b) \leftarrow \min_{x_a \in L} \left(\frac{\rho_{ab}}{\rho_a} \left(\theta_a(x_a) + \sum_{c \in \mathcal{N}(a)} m_{ca}(x_a) \right) - m_{ba}(x_a) + \theta_{ab}(x_a, x_b) \right)$$

In TRW, the graph is decomposed to a set of spanning trees. This decomposition is reflected via the probability $\frac{\rho_{ab}}{\rho_a}$. The probability is defined by a chance that a tree (chosen randomly) contains edge ab given that it contains a. Note that, in BP, this probability is simply set to 1. An interesting feature of the TRW algorithm is that it computes a lower bound on the energy (thus, related to the dual of LP). However, the seminal TRW algorithm [110] does not guarantee monotonic increase of the lower bound or convergence. This problem has been addressed by the extended tree-reweighted sequence (TRW-S) algorithm [54], which results in certain convergence properties. The TRW-S defines an arbitrary pixel ordering function $S(a)$. The messages are updated in order of increasing $S(a)$, and at the next iteration, are updated in the reverse order. At every iteration, the label configuration is done for every node by going through pixels in the order $S(a)$ and choose a label $x_a \in L$ such that:

$$\bar{x}_a = \arg\min_{x_a \in L} \quad \theta_a(x_a) + \sum_{S(b)<S(a)} \theta_{ab}(x_a, x_b) + \sum_{S(b)>S(a)} m_{ba}(x_a)$$

The labelling update based on this heuristic rule does not guarantee monotonic decrease of energy function (the primal objective function (4.5)) but only ensures that the lower bound (the dual objective) does not decrease. In practice, one could keep track of the lowest energy so far to determine the solution corresponding to the lowest energy.



(a) Non-convex function (b) Convex relaxation

Figure 4.9: *Convex relaxation of non-convex function. Non-convex function contains several local minima, the approximated convex function only has one global minimum.*

4.2.3 Linear programming relaxation

In the last few years, many studies have considered the relationship between combinatorial methods and continuous optimisation to establish the convergence properties of the combinatorial methods. Most examine the Linear Programming Relaxation to the MRF model, as the tightest relaxation within the class of convex relaxations. It is well-known that the MRF minimisation problem (4.5) is NP-hard and the objective function is highly-noncovex. In order to solve the problem, several aforementioned combinatorial methods have been proposed for approximating the solutions. However, combinatorial methods cannot guarantee convergence to optimality. Indeed, there are established examples that graph-cut or message-passing techniques fail to produce a solution [54, 58]. Non-convex problems may have many local minima. It is possible to overcome this limitation by applying convex relaxation techniques in order to obtain an approximate convex model that has a unique (global) minimum. Figure 4.9 illustrates a convex relaxation of a non-convex function. Convex relaxation is a common approximation technique to transform a problem into a convex domain and is a powerful setting for continuous optimisation. It has strong global convergence properties with numerous efficient solvers. Several approximation algorithms have been proposed in the literature such as linear programming relaxation [23, 110], quadratic programming relaxation [87] and second order cone programming relaxation [71, 61]. Amongst these relaxations, Kumar et. al. [60] have shown that LP relaxation provides the tightest bound to discrete MRF minimisation. There are numerous efficient algorithms for the LP relaxation of MRF, with established convergence properties. The LP relaxations of MRFs are based on the linear integer programming formulation. Let us consider the following binary variables and potential function:

- $x_{a,l} \in \{0,1\}$ and $x_{a,l} = 1 \iff$ label $l \in L$ is assigned to node $a \in V$.
- $x_{ab,lk} \in \{0,1\}$ and $x_{ab,lk} = 1 \iff$ label l, k are assigned to a pair of nodes $a, b \in V$ respectively; and $(a,b) \in E$.
- $\theta_{a,l} \overset{\text{def}}{=} \theta_a(x_a = l)$ is a unary cost of setting label l to node a.
- $\theta_{ab,lk} \overset{\text{def}}{=} \theta_{ab}(x_a = l, x_b = k)$ is a pairwise cost of assigning label l to node a and label k to node b.

Then the following integer programming (IP) can be shown to be equivalent to the task of minimising the MRF energy (4.5) [23, 113]:

$$\underset{x}{\text{minimise}} \quad \langle\theta, x\rangle = \sum_{a\in V}\sum_{l\in L}\theta_{a,l}.x_{a,l} + \sum_{ab\in E}\sum_{l,k\in L}\theta_{ab,lk}.x_{ab,lk} \tag{4.6a}$$

$$\text{such that} \quad x \in X^G = \left\{ x \;\middle|\; \begin{array}{cc} \sum_{l\in L} x_{a,l} = 1, & \forall a \in V \\ \sum_{j\in L} x_{ab,lk} = x_{a,l}, & \forall ab \in E, \forall l \in L \\ x_{a,l} \in \{0,1\},\ \forall a \in V, & x_{ab,lk} \in \{0,1\}, \forall ab \in E \end{array} \right\} \tag{4.6b}$$

In the above formulation, θ and x are the full vectors that consist of all potential and binary variable terms. X^G is the feasible set, which was originally called the *marginal polytope* [110]. These encode the properties of the graph G and MRF model. In particular, the first set of constraints simply express the fact that each node must admit one unique label, while the second set of constraints enforce consistency between the unary variables $x_{a,l}, x_{b,k}$ and the pairwise variables $x_{ab,lk}$, since they ensure that if $x_{a,l} = x_{b,k} = 1$ then $x_{ab,lk} = 1$ as well. Problem (4.6) represents exactly the discrete MRF minimisation (4.5), and is known to be NP-hard. The simplest and most common relaxation is the relaxation of the binary constraints, i.e. setting $x_{a,l} \geq 0$ and $x_{ab,lk} \geq 0$. The relaxation reduces the IP to an LP, the most common optimisation problem, understood and solved by many efficient optimisation solvers. The resulting polytope with continuous constraints is known as the *local polytope* [110], which contains additional fractional vertices, see Figure 4.10. Unfortunately, the large dimensionality of the image processing problem is too expensive for standard LP solvers such as simplex or interior point algorithms. It is possible to have a graph with hundreds of thousands of nodes and edges and multiplying these with a finite number of labels will result in millions of unknown variables. Therefore, it is neccesary to efficiently exploit graph structures and MRF properties (e.g. utilising combina-

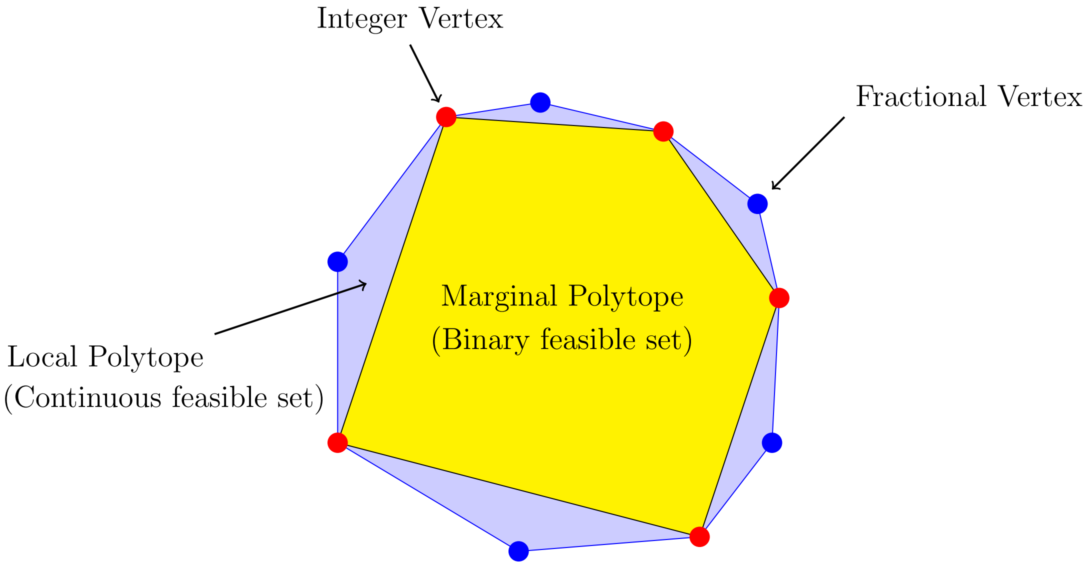


Figure 4.10: *Marginal polyopte only contains integer vertices, local polytope contains fractional vertices*

torial methods to solve the easy MRF problem in the inner loop [59], or using message-passing techniques in the iterative updates [110, 54]). For the ease of presentation, we will hereafter refer to (4.6) as the LP relaxation of MRF model (LP-MRF), subject to the *local polytope*, the original integer program of MRF models will be denoted as IP.

As already mentioned, TRW methods are tightly related to the LP-MRF (4.6). Based on the assumption that this relaxation provides a good approximation to the integer program [60], TRW methods hope to obtain an approximately optimal solution to the labelling problem, by solving the LP. However, TRW methods do not attempt to minimize the LP directly. Instead, they focus on solving the dual of that relaxation:

$$\max_{\theta \in C(\theta),\, \sum_{t \in T} \rho^t \theta^t = \theta} \sum_{t \in T} \rho^t \min_{x^t \in X^t} \langle \theta^t, x^t \rangle \tag{4.7}$$

In the LP dual formulation above, $C(\theta)$ is a convex combination of spanning trees, t denotes a tree contained in a tree set T that covers all edges and nodes at least once. X^t forms a *marginal polytope*, which is similar to X^G (4.6b), that reflects the structure of the corresponding tree $t \in T$. Dualising the LP provides the motivation underlying some other MAP estimation algorithms such as the max-sum diffusion algorithm [113, 30]. These methods operate on a dual of LP, and can essentially be understood as block coordinate ascent procedures applied to the dual. Obviously, the cost of any feasible solution to the dual LP yields a lower bound on the optimal MRF energy (i.e. the primal). Hence, solving the dual corresponds to a maximisation of this lower bound, which is essentially the key idea behind all the above mentioned techniques. Based on the dual solution, a solution to the original MRF IP can be extracted using basic heuristics or simple rounding procedures. Moreover, the quality of the resulting MRF solution depends critically on the quality of the estimated primal-dual gap (i.e, how large that gap is). However, none of the combinatorial methods can be guaranteed to achieve the optimal primal-dual gap (i.e. the gap goes to zero). In fact, as shown in [54], there exist examples that illustrate this point.

In order to address these limitations, Komodakis et. al. [59] propose a dual decomposion scheme to tackle the LP problem. The resulting optimisation is a generalisation of the dual problem (4.7), which can be shown to converge globally. Sontag et. al. [97] have shown that specialised algorithms for dual LP are equivalent to dual decomposion technique. In this thesis, we adopt a complete optimisation point of view and employ the dual decomposion technique in [58], that transforms the LP-MRF (4.6) to the maximisation of a large scale convex nondifferentiable function. There are numerous approaches for solving the latter including the seminal works of Nemirovski [48] and Nesterov [78] on First Order Methods (FOM). Recently, there have been further studies of FOM for the MRF problem. Following the projected-subgradient method [59], Jancsary et. al. [44] apply the incremental implementation, which is indeed a special case of [58]. Jojic et. al. [46] and Savchynskyy et. al. [92] employ the smoothing technique of Nesterov [78] to accelerate the theoretical convergence rate. Smoothing techniques benefit from fast convergence rate. However, the global solution is only a smooth-approximation of the lower bound. In addition, it is important to provide a good smoothing parameter to obtain a corresponding suboptimal approximation. Recently, Luong et. al. [66] proposed a method to solve the non-smooth optimisation problem of MRF using the weighted nonlinear projection method. This method computes optimal entropy projection updates at early iterations, before switching to standard subgradient updates when the entropy projection updates have become stable. The method sharpens the convergence result of the standard projected subgradient and has exhibited promising experimental results. Ravikumar et. al. [86] have also proposed a method based on cyclic entropy projection. However, their method operates on the primal domain, i.e. the LP-MRF (4.6), and has to perform an inner loop for every entropy projection. The inner loop is not guaranteed to converge and may contain excessively number of iterations.

4.3 Dual decomposition of LP-MRF

This section introduces the dual decomposition approach, also known as Lagrangian relaxation. With dual decomposition, the original (master) problem is decomposed into smaller (slave) subproblems with additional Lagrange multipliers. There are various ways to define subproblems, and the only requirement for a choice of decomposition is that a subproblem has to be a MRF problem that can be solved exactly and efficiently. In this thesis, we focus on acyclic structure MRF to define subproblems, i.e. a MRF problem on a graph without loop. It is well known that a MRF problem on a graph without loop can be solved within one iteration by the Belief Propagation algorithm [84]. The decomposition is subsequently optimised with respect to the Lagrange multipliers by the master problem, which acts as a coordinator between the subproblems, to encourage them to agree about the variables they share. Hereafter, we will use subproblem and slave interchangeably reagarding the MRF subproblem, and acyclic graph and tree are equivalently used for describing a graph without loop.

Tree decomposition. To this end, let T be a set of trees of the original graph G. The only requirement for the set T is that all trees (together) must cover every node and edge of graph G. For examples, T can be a collection of spanning trees where all trees (together) must cover every edge at least once, and each node is covered $|T|$ times, where $|T|$ is the number of trees in the set. A good tree decomposition for MRF subproblems is an interesting research topic, which has been studied by a number of authors [2, 54, 43, 45]. A simple example, and the one that we use in this thesis, is a collection of horizontal and vertical edges. This simple decomposition technique is popular for developing specialised algorithms to improve theoretical convergence properties from an optimisation point of view [92, 46, 72, 44]. By such a construction, each edge $ab \in E$ is covered exactly once by the set T and each node $a \in V$ appears twice, once in the horizontal tree and once in the vertical tree. Note that, for the 3D image problem, we have an additional tree that represents the depth, thus each node will appear three times in the 3D tree set. We do not consider the 3D case in this thesis, however the same framework for the 2D problem can be applied to the 3D problem easily. For each tree $t \in T$, there is a MRF problem defined only on the nodes and edges of the tree, which contains a vector of MRF potentials θ^t, as well as a vector of MRF variables x^t. These vectors have similar structures to the original MRF terms θ and x, except that the horizontal tree does not contain any terms that related to vertical edges and vice versa. As every edge is independently managed by one tree, there is no additional parameterisation needed for the edges, i.e. the edge terms $\theta_{ab,lk}$ and $x_{ab,lk}$ are distributed into either the horizontal tree or the vertical tree accordingly. However, each node $a \in V$ is shared by two trees, thus we need to define additional parameters for the tree nodes: $\theta^t_{a,l}$ and $x^t_{a,l}$, which can be vectorised into tree node vectors, denoted by θ^t and x^t. When the labellings (i.e. the solution $\bar{x}^t$) of two trees are combined into the original graph, it should hold that the labellings of the two trees are the same, and equal to the original graph labelling. To enforce this, the following constraint on the tree nodes is imposed:

$$x^t_{a,l} = x_{a,l}, \quad \forall a \in V, \ \forall l \in L$$

In addition, the unary potentials of the two trees need to preserve the original energy, by satisfying the following condition:

$$\sum_{t \in T} \theta^t_{a,l} = \theta_{a,l}, \quad \forall a \in V, \ \forall l \in L$$

The constraints are sometimes written in vectorised form as $x^t = x$ or $\sum\limits_{t \in T} \theta^t = \theta$. For ease of presentation, the *index* set is defined:

$$\mathcal{I} = \{ i \overset{\text{def}}{=} (a, l), \quad \forall a \in V, \ \forall l \in L \} \tag{4.8}$$

The tree unary potentials $\{\theta^t\}$ are defined prior to the optimisation process, thus they act as constants in the optimisation problem. Now, we have the following equivalence of original MRF problem:

$$\begin{aligned}
\min_{x \in X^G} \langle \theta, x \rangle \quad &\equiv \quad \begin{cases} \min\limits_{x^t, x \in X^G} & \sum\limits_{t \in T} \langle \theta^t, x^t \rangle \\ \text{such that} & x^t \in X^t, \quad \forall t \in T \\ & x^t = x, \quad \forall t \in T \end{cases} \\
&\equiv \quad \begin{cases} \min\limits_{x \in X^G} & \sum\limits_{t \in T} \min\limits_{x^t \in X^t} \langle \theta^t, x^t \rangle \\ \text{such that} & x^t = x, \qquad \forall t \in T \end{cases}
\end{aligned} \tag{4.9}$$

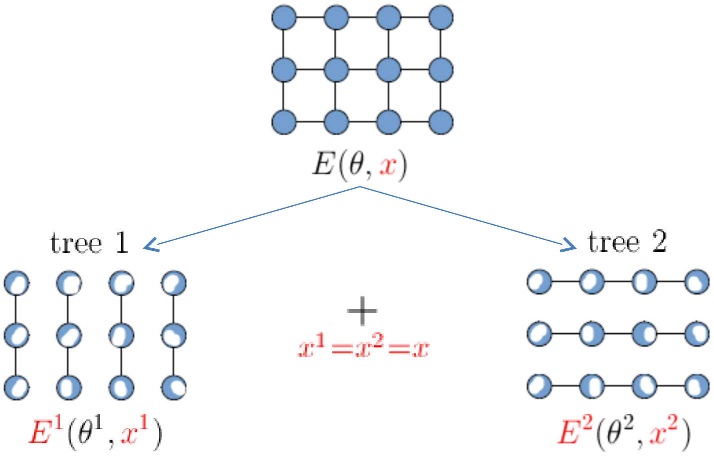


Figure 4.11: *MRF decomposition: the incomplete filled nodes in each tree represents its partial potential energies satisfying: $\theta^1 + \theta^2 = \theta$. The variables to optimised are x^1, x^2, x where x^1, x^2 can be optimised independently in E^1, E^2 if the coupling constraints $x^1 = x^2 = x$ are omitted.*

Clearly, each summand $\langle\theta^t, x^t\rangle$ represents a tree-MRF problem, and can be optimised independently thereby obtaining (4.9). Let $E(\theta, x)$ denotes the minimum energy of any arbitrary MRF model, e.g. if $x \in X^G$ then $E(\theta, x)$ represents the minimum potential energy of the original MRF model; if $x^t \in X^T$ then $E^t(\theta^t, x^t)$ is the minimum potential energy of the tree MRF problem. This notation reduces the the form of (4.9) to:

$$E(\theta, x) = \min_{x \in X^G} \langle\theta, x\rangle = \min_{x \in X^G, \forall t \in T : x^t = x} \sum_{t \in T} E^t(\theta^t, x^t)$$

Figure 4.11 illustrates the above decomposion problem. It is clear that the coupling constraints $x^t = x$ make the optimisation problem difficult. Without it, one could optimise each small MRF problems (one per tree $t \in T$) independently. Therefore, it is natural to relax these coupling constraints via the Lagrangian dual function as:

$$\begin{aligned} E(\{\lambda^t, x^t\}) &= \min_{x \in X^G} \sum_{t \in T} \min_{x^t \in X^t} \langle\theta^t, x^t\rangle + \sum_{t \in T} \lambda^t.(x^t - x) \qquad (4.10) \\ &= \sum_{t \in T} \min_{x^t \in X^t} \langle\theta^t + \lambda^t, x^t\rangle - \min_{x \in X^G} \left(\sum_{t \in T} \lambda^t\right).x \\ &= \begin{cases} \sum_{t \in T} \min_{x^t \in X^t} \langle\theta^t + \lambda^t, x^t\rangle & , \text{ if } \sum_{t \in T} \lambda^t = 0 \\ -\infty & , \text{ if } \sum_{t \in T} \lambda^t \neq 0 \end{cases} \end{aligned}$$

Clearly, we omit the case where the energy is $-\infty$ and each summand $\langle\theta^t + \lambda^t, x^t\rangle$ is optimised

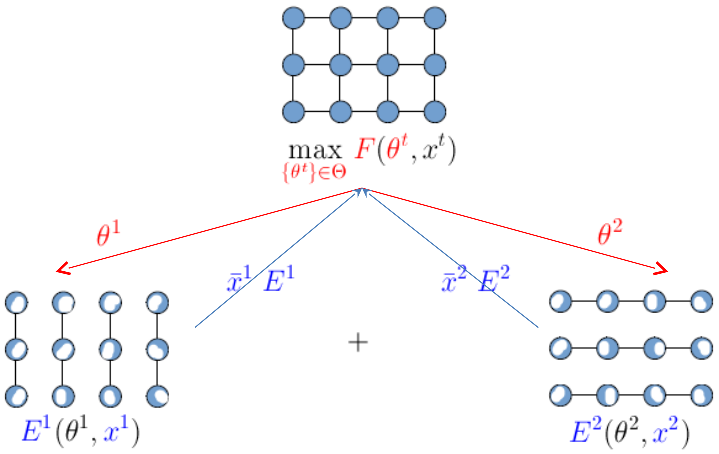


Figure 4.12: *MRF dual decomposition: let $F(\theta^t, x^t) = E^1(\theta^1, x^1) + E^2(\theta^2, x^2)$. At each iteration, the master problem distributes potential energies θ^1, θ^2 to the two MRF trees. Each MRF tree is optimised by BP and returns information about its optimal energies E^1, E^2 and labelling x^1, x^2. The information is passed to the master problem to coordinate the next distribution of potential energies.*

independently thereby obtaining the final equality. Via the dual problem, we have eliminated the coupled constraints $x^t = x$, and obtained a new objective function which is a sum over simple MRF problems (where smaller MRFs are defined on the trees). Thus, we now can set up a dual problem, i.e. maximise the above dual function $E(\{\lambda^t, x^t\})$:

$$\begin{aligned} \max_{\lambda^t} \quad & \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t + \lambda^t, x^t \rangle \\ \text{such that} \quad & \textstyle\sum_{t\in T} \lambda^t = 0 \end{aligned} \tag{4.11}$$

In order to avoid ambiguity, let the symbol $\hat{\theta}$ denotes the (constant) original unary potentials. Given that the following conditions always satisfied:

$$\begin{cases} \text{Predefined distribution:} & \sum_{t\in T} \theta^t = \hat{\theta} \\ \text{Dual fesibility:} & \sum_{t\in T} \lambda^t = 0 \end{cases}$$

The reparameterisation $\theta^t \overset{\text{def}}{=} \theta^t + \lambda^t$ leads to an equivalent optimisation problem to (4.11):

$$\max_{\theta\in\Theta} \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t, x^t \rangle = \max_{\theta\in\Theta} \sum_{t\in T} E^t(\theta^t, x^t) \tag{4.12}$$

where

$$\Theta = \left\{ \{\theta^t\} \middle| \sum_{t\in T} \theta^t = \hat{\theta} \right\}$$

Problem (4.12) encodes the Lagrange multipliers into the energy potentials, and optimises over the new potential terms, instead of the multipliers. The dual problem can be referred to a master problem, while each slave problem $E^t(\theta^t, x^t)$ simply amounts to optimising an MRF over a tree $t \in T$. In this setting, at every iteration, the slave MRF returns information about the objective function E^t and the labelling x^t. The master problem takes this information and redistributes the node potential energies θ^t to each tree t in order to encourage increased labelling agreement between the trees. This process is repeated until no further improvement can be made on the label agreement. Figure 4.12 illustrates one iteration of the process. Problem (4.12) can be shown to be equivalent to the LP relaxation of the discrete MRF interger problem (4.6a). The primal-dual equivalence via weak duality is known in the optimisation literature [9]. For the completeness of the thesis, a concise proof of the primal-dual equivalence for MRFs follows.

Theorem 4.3.1 *Problem* (4.12) *is equivalent to the LP problem* (4.6a) *where* X^G *is a local polytope (with relaxed constraints* $x \geq 0$*).*

Proof Via reparameterisation and elimination of the coupling constraints $x^t = x$, we obtain problem (4.12). Thus, we have:

$$
\begin{aligned}
\max_{\theta\in\Theta} \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t, x^t\rangle &= \max_{\sum_{t\in T}\lambda^t=0} \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t + \lambda^t, x^t\rangle \\
&= \max_{\lambda^t} \min_{x\in X^G} \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t, x^t\rangle + \sum_{t\in T} \lambda^t.(x^t - x) \\
&= \max_{\lambda^t} \min_{x\in X^G,\{x^t\in X^t\}} \sum_{t\in T} \langle \theta^t, x^t\rangle + \sum_{t\in T} \lambda^t.(x^t - x) \\
&= \min_{x\in X^G,\{x^t\in X^t\}} \sum_{t\in T} \langle \theta^t, x^t\rangle + \max_{\lambda^t} \sum_{t\in T} \lambda^t.(x^t - x) \\
&= \begin{cases} \min\limits_{x\in X^G,\{x^t\in X^t\}} \sum\limits_{t\in T} \langle \theta^t, x^t\rangle \quad , & \text{if} \quad x^t = x,\ \forall t \in T \\ \infty \quad , & \text{if} \quad x^t \neq x,\ \forall t \in T \end{cases} \\
&= \min_{x\in X^G,\{x^t\in X^t\},\{x^t=x\}} \sum_{t\in T} \langle \theta^t, x^t\rangle \\
&= \min_{x\in X^G,\{x^t\in X^t\},\{x^t=x\}} \left\langle \sum_{t\in T} \theta^t, x \right\rangle \qquad (4.13) \\
&= \min_{x\in X^G,\{x^t\in X^t\},\{x^t=x\}} \langle \hat{\theta}, x\rangle = \min_{x\in X^G} \langle \hat{\theta}, x\rangle
\end{aligned}
$$

The equality (4.13) comes from the fact that $\forall t \in T : x^t = x$, therefore every x^t cab be replaced with x accordingly. The final equality comes from the predefined distribution $\sum_{t \in T} \theta^t = \hat{\theta}$. In addition, the convex hull of all tree set $\{X^t\}$ coincides with the local marginal polytope X^G, therefore the constraints $\{x^t \in X^t\}, \{x^t = x\}$ become redundant and can be omitted. ∎

Remarks. At this point, there could be a false impression that theorem 4.3.1 only holds for horizontal and vertical tree decomposition. However, the theorem is much more general than that, and it is true as long as there exists a global optimiser for each subproblem.

Chapter 5

First Order Methods for LP-MRF

The primary goal of this chapter is to derive efficient optimisation methods for solving the large scale LP-MRF (4.6a). A common technique is to use polynomial time interior point methods [80], that can produce high accuracy solutions in limited number of iterations. However, all interior point methods share a common drawback: the computational effort per iteration grows rapidly with problem dimensionality. As a result, polynomial time methods eventually become intractable, i.e. for a large scale problem, a single iteration may take a long time to compute. Unfortunately, large scale problems often arise in image processing, in particular, for the LP-MRF. In this chapter we propose a first order *subgradient* method (FOM) that takes into account of the problem structure. In the final section of the previous chapter, we showed that tree-structure MRFs can be solved exactly and efficiently by the belief-propagation method. In this chapter, the use of the dual decomposition technique is proposed to reformulate the LP-MRF as a dual LP-MRF. It was also shown that, the LP-MRF and dual LP-MRF are equivalent. Therefore, solving a dual LP-MRF is sufficient to derive a solution of LP-MRF. Interestingly, the dual LP-MRF belongs to a class of nonsmooth convex optimisation problems. In this chapter, we utilise FOM to solve the dual LP-MRF (4.12):

$$\max_{\theta \in \Theta} \sum_{t \in T} E^t(\theta^t, x^t)$$

which belongs the a class of nonsmooth optimisation:

$$\min_{x \in C} F(x)$$

where $F(x)$ is a nondifferentiable function and C is a simple convex set. One disadvantage of FOM is that only a sublinear convergence rate is guaranteed, i.e. it requires $\mathrm{O}(\frac{1}{\epsilon^2})$ iterations to obtain a solution within $\mathrm{O}(\epsilon)$ optimality. However, FOM benefits from low computational cost per iteration. This is needed in large scale problems. In addition, for problems with favourable geometry, a good FOM exhibits near dimensionality independence with respect to its convergence properties. The computer memory requirement for FOM can be much smaller than interior point methods for solving the primal problem, which is also an advantage for large memory needed for image problems. In this chapter, we describe one standard FOM, the *subgradient* projection method, before developing a novel nonlinear weighted projection method to accelerate the performance. Competitive theoretical properties of the methods are analysed and supported by experimental results.

5.1 Projected subgradient

One typical FOM is the subgradient projection method. In this section, we apply a dual decomposition technique and use FOM with subgradient projection [9, Section 6.4] to develop a simple and efficient algorithm for our large scale problem. The work in this chapter addresses the nondifferentiable dual LP-MRF probem:

$$\max_{\theta\in\Theta} F(\theta) = \max_{\theta\in\Theta} \sum_{t\in T} \min_{x^t\in X^t} \langle \theta^t, x^t\rangle = \max_{\theta\in\Theta} \sum_{t\in T} E^t(\theta^t, x^t) \tag{5.1a}$$

$$\text{where} \qquad \Theta = \left\{ \{\theta^t\} \middle| \sum_{t\in T} \theta^t = \hat{\theta} \right\} \equiv \left\{ \{\theta^t_i\} \middle| \sum_{t\in T} \theta^t_i = \hat{\theta}_i,\ \forall i \in \mathcal{I} \right\} \tag{5.1b}$$

the potential preservation set Θ is an union of $|V| \times |L|$ disjoint sets Θ_i associated with each index $i \in \mathcal{I} = \{(a,l), \quad \forall a \in V,\ \forall l \in L\}$, i.e. $\Theta =_{\otimes} \Theta_i\ \forall i \in \mathcal{I}$, where:

$$\Theta_i = \left\{ \theta^t_i \middle| \sum_{t\in T} \theta^t_i = \hat{\theta}_i \right\} \tag{5.1c}$$

The function $F(\theta)$ is concave and nonsmooth (maximising a concave function is a convex optimisation problem), therefore we use subgradients of the concave function F in our algorithms. The subgradient method is specified in Algorithm 5.1 and is developed in detail below.

Definition 5.1.1 *A vector $F'(\theta) \in \mathbb{R}^n$ is a* subgradient *of $F : \mathbb{R}^n \to \mathbb{R}$ at $\theta \in$ Domain(F), if $\forall \gamma \in$ Domain(F),*

$$F(\gamma) \leq F(\theta) + \langle F'(\theta), \gamma - \theta \rangle \tag{5.2}$$

Then, a projected subgradient method uses a simple update rule for the $(k+1)^{th}$ iteration:

$$\theta^{(k+1)} = \pi_\Theta \left(\theta^{(k)} + \tau_k F'(\theta^{(k)}) \right) \tag{5.3}$$

In the update (5.3), τ_k denotes a positive multiplier, $\pi_\Theta(.)$ denotes the projection on to the set Θ, while $F'(\theta^{(k)})$ represents a subgradient of $F(.)$ at $\theta^{(k)}$. In the projected subgradient method, there is no restriction on choosing a subgradient. Here, a subgradient based on information given by the slave problems, $E^t(\theta^t, x^t) = \min_{x^t \in X^t} \langle \theta^t, x^t \rangle$, is used. Note that, $F(\theta) \equiv F(\{\theta^t\})$ is defined as the linear combination of the functions corresponding to the disjoint trees. Therefore, its subgradients (corresponding to each tree) can be computed independently.

Lemma 5.1.2 *A subgradient of $F(\theta)$ can be chosen by:*

$$F'(\theta) \overset{\text{def}}{=} \bar{x} \tag{5.4a}$$

$$\text{where} \quad \bar{x} \overset{\text{def}}{=} \{\bar{x}^t\}, \quad \forall t \in T : \quad \bar{x}^t = \arg \min_{x^t \in X^t} \langle \theta^t, x^t \rangle \tag{5.4b}$$

Proof From the definition (5.4b), we know $\bar{x}^t$ is not the optimal solution of $\min_{x^t \in X^t} \langle \gamma^t, x^t \rangle$, i.e.

$$\forall t \in T : \quad E^t(\gamma^t, x^t) \overset{\text{def}}{=} \min_{x^t \in X^t} \langle \gamma^t, x^t \rangle \leq \langle \gamma^t, \bar{x}^t \rangle$$

In addition,

$$F(\gamma) = \sum_{t \in T} E^t(\gamma^t, x^t) \leq \sum_{t \in T} \langle \gamma^t, \bar{x}^t \rangle = \sum_{t \in T} \langle \theta^t, \bar{x}^t \rangle + \langle \gamma^t - \theta^t, \bar{x}^t \rangle = \sum_{t \in T} E^t(\theta^t, x^t) + \langle \gamma^t - \theta^t, \bar{x}^t \rangle$$

Therefore, $\quad F(\gamma) \leq F(\theta) + \langle F'(\theta), \gamma - \theta \rangle$ where $F'(\theta) = \bar{x}$. ∎

Using the chosen subgradients and solving the Lagrangian of (5.1a), a simple update rule for the projected subgradient iteration is obtained.

Proposition 5.1.3 *The projected subgradient update* (5.3) *for the dual LP-MRF* (5.1a) *is given by:*

$$\forall\, t \in T,\, \forall i \in \mathcal{I}: \quad \theta_i^{t(k+1)} = \theta_i^{t(k)} + \tau_k \left(\bar{x}_i^{t(k)} - \frac{\sum_{t\in T} \bar{x}_i^{t(k)}}{|T|} \right) \tag{5.5}$$

Proof It is well-known that the projected subgradient update (5.3) comes from the proximal iteration [9]:

$$\begin{aligned}
\theta^{(k+1)} &= \pi_\Theta\left(\theta^{(k)} + \tau_k F'(\theta^{(k)})\right) \equiv \arg\max_{\theta\in\Theta} \left\langle \theta, F'(\theta^{(k)})\right\rangle - \frac{1}{2\tau_k}\left\|\theta - \theta^{(k)}\right\|_2^2 \\
&= \arg\max_{\theta\in\Theta} \sum_{t\in T}\sum_{i\in\mathcal{I}} \theta_i^t . F'(\theta_i^{t(k)}) - \frac{1}{2\tau_k}(\theta_i^t - \theta_i^{t(k)})^2 \\
&= \arg\max_{\forall i\in\mathcal{I}:\, \sum_{t\in T}\theta_i^t = \hat{\theta}_i} \sum_{t\in T}\sum_{i\in\mathcal{I}} \theta_i^t . \bar{x}_i^{t(k)} - \frac{1}{2\tau_k}(\theta_i^t - \theta_i^{t(k)})^2
\end{aligned}$$

The Lagrangian is given by:

$$\begin{aligned}
L(\theta,\lambda) &= \sum_{i\in\mathcal{I}}\left[\sum_{t\in T}\left(\theta_i^t.\bar{x}_i^{t(k)} - \frac{1}{2\tau_k}(\theta_i^t - \theta_i^{t(k)})^2\right) + \lambda_i\left(\sum_{t\in T}\theta_i^t - \hat{\theta}_i\right)\right] \\
\nabla_{\theta_i^t} L &= \bar{x}_i^{t(k)} - \frac{1}{\tau_k}(\theta_i^t - \theta_i^{t(k)}) + \lambda_i = 0 \\
\Rightarrow \theta_i^t &= \theta_i^{t(k)} + \tau_k(\bar{x}_i^{t(k)} + \lambda_i) && \text{(5.6a)} \\
\nabla_{\lambda_i} L &= \sum_{t\in T}\theta_i^t - \hat{\theta}_i = \sum_{t\in T}\theta_i^{t(k)} + \tau_k(\bar{x}_i^{t(k)} + \lambda_i) - \hat{\theta}_i = 0 \\
\Rightarrow \lambda_i &= \frac{-\sum_{t\in T}\bar{x}_i^{t(k)}}{|T|} && \text{(5.6b)} \\
\xrightarrow{(5.6a),(5.6b)} \theta_i^{t(k+1)} &= \theta_i^{t(k)} + \tau_k\left(\bar{x}_i^{t(k)} - \frac{\sum_{t\in T}\bar{x}_i^{t(k)}}{|T|}\right)
\end{aligned}$$

■

The subgradient update (5.5) requires very few computations per iteration due to the use of subgradients, as indicated by the following corollaries:

Corollary 5.1.4 *The projected subgradient updates are only required at the nodes which are not assigned the same label by all trees (disagreement nodes).*

For instance, the node $\alpha \in V$ is assigned the same label $l_\alpha \in L$ by all trees, i.e.

$$x^{t(k)}_{\alpha,l\neq l_\alpha} = 0 \quad \text{and} \quad x^{t(k)}_{\alpha,l_\alpha} = 1, \quad \forall t \in T$$

Clearly, at the node α, the trees need to update the node's potential by:

$$\forall t \in T,\ \forall l \in L : \quad \theta^{t(k+1)}_{\alpha,l} = \theta^{t(k)}_{\alpha,l} + \tau_k \left(\bar{x}^{t(k)}_{\alpha,l} - \frac{\sum_{t\in T} \bar{x}^{t(k)}_{\alpha,l}}{|T|} \right)$$

Now:

$$\begin{array}{ll} \text{For } l = l_\alpha : & \bar{x}^{t(k)}_{\alpha,l_\alpha} - \frac{\sum_{t\in T} \bar{x}^{t(k)}_{\alpha,l_\alpha}}{|T|} = 1 - \frac{|T|}{|T|} = 0 \\ \text{For } l \neq l_\alpha : & \bar{x}^{t(k)}_{\alpha,l\neq l_\alpha} - \frac{\sum_{t\in T} \bar{x}^{t(k)}_{\alpha,l\neq l_\alpha}}{|T|} = 0 - \frac{0}{|T|} = 0 \end{array} \tag{5.7}$$

Corollary 5.1.5 *At the disagreement nodes, projected subgradient updates are only needed at disagreeing labels.*

The above results reduce further the computational cost per iteration. Let two trees $t_1, t_2 \in T$ cover the node $\alpha \in V$, and MRF slaves E^1, E^2 which assign labels l_1, l_2 respectively to that node. Then, it can be seen that the following potential updates will take place:

$$\theta^{t_1(k+1)}_{\alpha,l} = \begin{cases} \theta^{t_1(k)}_{\alpha,l} + \tau_k/2 & \text{if} \quad l = l_1 \\ \theta^{t_1(k)}_{\alpha,l} - \tau_k/2 & \text{if} \quad l = l_2 \\ \theta^{t_1(k)}_{\alpha,l} & \text{if} \quad l \neq l_1, l_2 \end{cases}, \quad \theta^{t_2(k+1)}_{\alpha,l} = \begin{cases} \theta^{t_2(k)}_{\alpha,l} - \tau_k/2 & \text{if} \quad l = l_1 \\ \theta^{t_2(k)}_{\alpha,l} + \tau_k/2 & \text{if} \quad l = l_2 \\ \theta^{t_2(k)}_{\alpha,l} & \text{if} \quad l \neq l_1, l_2 \end{cases}$$

This straightforwardly extends to cases where three or more trees cover a node. When two trees disagree over a node labelling, the updates only take places at disagreeing labels l_1 and l_2. The effect of this is that the master problem tries to readjust the potentials of a node with disagreement, so that a common label assignment to that node is more likely in the next iteration. At this point, it is also worth noting the difference between the subgradient method and the TRW [110]. While TRW uses the tree min-marginals in order to update the unary potentials θ^t, the subgradient method relies on the labelling of slave MRF ($\bar{x}^t$) for that task. Furthermore, the computational cost per iteration of the subgradient method is much lower than TRW as the algorithm converges towards an optimum. This is because the subgradient method only updates disagreeing nodes and labels, comapared to the TRW update of all dual variables.

Algorithm 5.1: MRF Subgradient method $x \leftarrow \texttt{SG}(\hat{\theta})$

```
Define a tree collection $T$ contains horizontal and vertical trees;
Define the index set $\mathcal{I}$ (4.8);
Set $\theta_i^{t(1)} = \hat{\theta}_i/|T|$;
for $k = 1, ..., K$ do
    for $t \in T$ do
        Compute $\bar{x}^{t(k)} \rightarrow \texttt{BeliefPropagation}(\theta^{t(k)})$
    Update primal $x^{(k)}$ by (5.8);
    Compute primal $E(\hat{\theta}, x^{(k)})$ and dual $F(\theta^{(k)})$;
    Compute step-size $\tau_k$ by (5.17);
    for $i \in \mathcal{I}$ do
        Let update $= \sum_{t \in T} \bar{x}_i^{t(k)}$;
        if $0 <$ update $< |T|$ then
            Update the potentials $\theta^{(k+1)}$ by (5.5)
Return $x^{(K)}$;
```

5.1.1 Computing the primal solutions

Once the dual solutions are computed, the primal solution needs to be estimated. This topic has attracted much attention in the optimisation literature. We utilise a popular procedure that recovers the primal solutions via ergodic sequences of the dual subgradients. Ergodic primal convergence analysis has been studied by many authors to bridge the primal-dual gap in linear programming [95, 96] and constrained convex optimisation [63]. One of the simplest primal approximation takes the weighted average of dual subgradients at the k^{th}, of the form:

$$x^{(k)} = \frac{\sum_{j=1}^{k} F'(\theta^{(k)})}{k} = \frac{\sum_{j=1}^{k} \sum_{t \in T} \bar{x}^{t(k)}}{k}$$

In the above, it is necessary to keep track of all historical subgradients, which may be limited by the amount of computer memory available. An efficient way to do this is to set up a counter for every node that counts the number of times a label is assigned to a node. The label with the biggest count at a current time is chosen to be the current label of the node, i.e.

$$\begin{array}{rlll} \forall i \in \mathcal{I}: & \text{counter}(i) & = & \text{counter}(i) + \sum_{t \in T} \bar{x}_i^{t(k)} \\ \forall a \in V: & l_a & = & \arg\max_l \{\text{counter}(a, l)\} \\ \text{Primal Solution:} & x_{a,l}^{(k)} & = & (l == l_a) \end{array} \tag{5.8}$$

where the index $i \equiv (a, l)$ can be extracted directly from the definition of the index set $\mathcal{I}$ (4.8). In this setting, we obtain a feasible but suboptimal MRF labelling. In practice, the number of feasible violations is very small and proportional to the number of inconsistent nodes, i.e. the nodes that disagree on a common label assignment. As we maximise the dual problem, we also minimise the number of nodes in disagreement. If there is no disagreement, we obtain the true optimal labelling. It also follows that the number of feasibility violations is related to the convergence properties of the primal solutions [75]. One important feature of computing primal solutions is the information it provides regarding the primal-dual gap. Using the primal-dual gap, we can develop an adaptive step-size that works much better than standard step-size strategies.

5.1.2 Convergence results

Much literature [9, 76, 81] exists on the convergence of the projected subgradient method. This section focuses on the nonasymtotic convergence rate, where the algorithm is guaranteed to converge within some range of the optimal value. It is well known that this range is a function of the number of iterations and is sensitive to the step-size strategy. In particular, we show that the subgradient method finds an $\mathrm{O}(\frac{1}{\sqrt{k}})$ suboptimal point within k iterations (in other words, ϵ-suboptimal point within $\mathrm{O}(\frac{1}{\epsilon^2})$ steps). The convergence proof relies on certain assumptions described in the following. These assumptions are simple but sufficiently general to cover most convex optimisation problems.

Assumption 5.1.6 *The following assumptions are often satisfied by many problems:*

- *Problem* (5.1a) *is solvable, i.e. there exist an optimal* $\theta^* = \arg\min_{\theta \in \Theta} F(\theta)$.
- *Each set* $\Theta_i \subset \mathbb{E}$, $\forall i \in \mathcal{I}$, *is a closed convex set in a finite dimensional Euclidean space* $\mathbb{E}$ *with a known upper bound* Ω_i *on the distance between the initial point to the optimal point:* $\Omega_i = \sup \frac{1}{2}\|\theta_i^{(1)} - \theta_i^*\|_2^2$, *then the domain bound is given by:*

$$\Omega = \sum_{i \in \mathcal{I}} \Omega_i \tag{5.9}$$

- *$F : \Theta \to \mathbb{R}$ is a $\mathcal{L}$-Lipschitz continuous convex function, i.e.*

$$|F(\theta) - F(\gamma)| \leq \mathcal{L}\|\theta - \gamma\|_2 \qquad \forall\, \theta, \gamma \in \Theta$$

the Lipschitz condition also implies that the subgradients of F are bounded by $\mathcal{L}$, i.e.

$$\mathcal{L} = \sup_{\theta \in \Theta} \|F'(\theta)\|_2 \tag{5.10}$$

Theorem 5.1.7 *Given that the assumptions 5.1.6 are satisfied and let $\{\theta^{(k)}\}$ be the sequence generated by* (5.3) *(explicitly by* (5.5)*), let $\bar{\theta} = \arg\max_{\{\theta^{(k)}\}} F(\theta^{(k)})$, then for any $k \geq 1$:*

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{\mathcal{L}\sqrt{2\Omega}}{\sqrt{K}} = \frac{\sqrt{\left(\sum_{i\in\mathcal{I}} \sup \|\theta_i^{(1)} - \theta_i^*\|_2^2\right)|V||T|}}{\sqrt{K}} \tag{5.11}$$

Proof

The key quantity in the proof is the distance to the optimal set. Let θ^* be the optimal point, we have:

$$\begin{aligned} \|\theta^{(k+1)} - \theta^*\|_2^2 &= \|\pi_\Theta\left(\theta^{(k)} + \tau_k F'(\theta^{(k)})\right) - \theta^*\|_2^2 \leq \|\theta^{(k)} + \tau_k F'(\theta^{(k)}) - \theta^*\|_2^2 & (5.12\text{a}) \\ &= \|\theta^{(k)} - \theta^*\|_2^2 + 2\tau_k \langle F'(\theta^{(k)}), \theta^{(k)} - \theta^* \rangle + \tau_k^2 \|F'(\theta^{(k)})\|_2^2 & \\ &\leq \|\theta^{(k)} - \theta^*\|_2^2 + 2\tau_k\left(F(\theta^{(k)}) - F(\theta^*)\right) + \tau_k^2\|F'(\theta^{(k)})\|_2^2 & (5.12\text{b}) \end{aligned}$$

The inequality in (5.12a) comes from the fact that a projected point is closer to any feasible point, and (5.12b) is based on the subgradient inequality of the concave function. Applying the above inequality recursively, we have:

$$\|\theta^{(k+1)} - \theta^*\|_2^2 \leq \|\theta^{(1)} - \theta^*\|_2^2 + 2\sum_{k=1}^{K} \tau_k\left(F(\theta^{(k)}) - F(\theta^*)\right) + \sum_{k=1}^{K} \tau_k^2 \|F'(\theta^{(k)})\|_2^2$$

By definition of the subgradient $F'(\theta^{(k)}) = \{\bar{x}^{(k)}\}$, where $\bar{x}_i^t \in \{0, 1\}$, the Lipschitz constant is equal to:

$$\mathcal{L} = \sup_{\theta\in\Theta} \|F'(\theta)\|_2 = \sqrt{|V||L||T|} = \sqrt{|\mathcal{I}||T|} \tag{5.13}$$

Using $\|\theta^{(k+1)} - \theta^*\|_2^2 \geq 0$ and $\|\theta^{(1)} - \theta^*\|_2^2 \leq 2\Omega$, we have:

$$\begin{array}{rrcl} & 2\sum_{k=1}^{K} \tau_k \left(F(\theta^*) - F(\theta^{(k)})\right) & \leq & 2\Omega + \mathcal{L}^2 \sum_{k=1}^{K} \tau_k^2 \\ \text{However,} & \forall k : F(\theta^*) - F(\bar{\theta}) & \leq & F(\theta^*) - F(\theta^{(k)}) \\ \text{So,} & F(\theta^*) - F(\bar{\theta}) & \leq & \dfrac{2\Omega + \mathcal{L}^2 \sum_{k=1}^{K} \tau_k^2}{2\sum_{k=1}^{K} \tau_k} \end{array} \tag{5.14}$$

The suboptimality bound depends on the sequence of step-size $\{\tau_k\}$, based on the assumption that the upper bounds Ω and $\mathcal{L}$ are known. One therefore seeks the best sequence $\{\tau_k\}$ which minimises the bound (5.14). This bound is a convex function of $\{\tau_k\}$ and minimising the RHS of (5.14) with respect to $\{\tau_k\}$ reduces the suboptimality to:

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{\mathcal{L}\sqrt{2\Omega}}{\sqrt{K}} \tag{5.15}$$

at the optimal stepsize:

$$\forall k = 1, .., K : \tau_k = \tau = \frac{\sqrt{2\Omega}}{\mathcal{L}\sqrt{K}} \tag{5.16}$$

Using assumption 5.1.6, we have:

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{\sqrt{\left(\sum_{i \in \mathcal{I}} \sup \|\theta_i^{(1)} - \theta_i^*\|_2^2\right) |\mathcal{I}||T|}}{\sqrt{K}}$$

∎

Remarks. The optimal step-size strategy $\tau = \frac{\sqrt{2\Omega}}{\mathcal{L}\sqrt{K}}$ requires the value of Ω, to be known a priori. This information may not always be available. However, there are various step-size strategies which guarantee convergence of the optimality bound (5.14) that do not require prior knowledge. For example:

- Constant step-size: setting $\tau_k = \tau$, gives:

  $$F(\theta^*) - F(\bar{\theta}) \leq \frac{2\Omega + K\mathcal{L}^2\tau^2}{2K\tau}$$

  The optimality bound converges to $\mathcal{L}^2\tau/2$ as $K \to \infty$. If Ω is known, then τ can be set to the optimal step-size (5.16) to obtain the optimal bound.

- Normalised subgradient with constant step length: set $\tau_k = \eta/\|F'(\theta^{(k)})\|_2$, thus $\tau_k \geq \eta/\mathcal{L}$ and the suboptimality bound becomes:

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{2\Omega + K\eta^2}{2\sum_{k=1}^{K}\tau_k} \leq \frac{2\Omega + K\eta^2}{2K\eta/\mathcal{L}}$$

The RHS converges to $\mathcal{L}\eta/2$ as $K \to \infty$. This is a general idea for subgradient methods; however, in our choice of subgradient, the norm over the subgradients is a constant (5.13) regardless of the value of the subgradients (the subgradients always contain the same number of 0s and 1s, the difference being their positions). Thus, the strategy for our problem is basically similar to that of constant step-size, with addition of gradient normalisation.

- Square summable but not summable: for example, let $\tau_k = \eta/k$, then we have:

$$\sum_{k=1}^{\infty}\tau_k^2 < \infty, \qquad \sum_{k=1}^{\infty}\tau_k = \infty$$

then as $k \to \infty$, the numerator of (5.14) converges while the denominator diverges, thus the bound converges.

- Diminishing step-size: if the sequence $\{\tau_k\}$ converges to zero and is nonsummable, eg. $\tau_k = \eta/k$, then the RHS of (5.14) also converges to zero [9].

- Normalised gradient with square summable but not summable step length: using the same properties as above. Now let $\tau_k = \eta_k/\|F'(\theta^{(k)}\|_2$, where (for example) $\eta_k = 1/k$. Then the suboptimality bound reduces to:

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{2\Omega + \sum_{k=1}^{K}\eta_k^2}{2\sum_{k=1}^{K}\tau_k} \leq \frac{2\Omega + \sum_{k=1}^{K}\eta_k^2}{2/\mathcal{L}\sum_{k=1}^{K}\eta_k}$$

As $K \to \infty$, the RHS converges to zero.

5.1.3 Speeding up the subgradient methods

The above step-size rules are based on a deterministic choice, where the sequence of step-sizes needs to be defined prior to gradient updates. On the other hand, one could use the information about current and historical information in order to adjust the step-size. This type of step-size can be regarded as an adaptive or dynamic update. Interestingly, the key quantity to proving convergence of the subgradient method is the distance Ω from the optimal solution. Furthermore, this distance is proportional to the primal-dual gap, i.e. $E(\hat{\theta}, x^{(k)}) - F(\theta^{(k)})$; and it is related directly to the optimal step-size strategy $\tau = \frac{\sqrt{2\Omega}}{\mathcal{L}\sqrt{K}}$. This motivates an adaptive step-size strategy for the projected subgradient method:

$$\tau_k = \frac{|E(\hat{\theta}, x^{(k)}) - F(\theta^{(k)})|}{|\mathcal{I}||T|} \tag{5.17}$$

The absolute value for the primal-dual gap is used as it is possible (but uncommon) that subgradient methods may lead to outliers that make the dual greater than the estimated primal. Using a dynamic step-size can be justified theoretically, by noting that as $k \to \infty$, the primal-dual gap $\to 0$. The defined sequence is therefore diminishing. From the MRF point of view, at early iterations when there are still many inconsistent node labellings, large changes need to be made on the nodes' potentials. As the primal-dual gap becomes smaller, less modifications is needed.

Other methods for improving speed include the cutting plane method, ellipsoid method, conic combination of previous subgradients [9, 81]. While these methods generally improve the convergence rate, they suffer from expensive computational cost per iteration. In the particular case of tree-labelling, subgradient updates occur only at a few disagreeing nodes and labels. However, if the direction of search changes to be used in complex algorithms, it may lead to updates being required everywhere. In the next section, a method is proposed to improve the convergence rate while keeping the inexpensive computational cost per iteration.

5.2 Nonlinear projection (Mirror Descent)

Projected subgradient methods provide a very efficient update per iteration for the MRF dual problem. However, they suffer from slow convergence rate and sensitivity to step-sizes. Although the primal-dual gap is used as a mechanism to dynamically adjust the step-size, no improved convergence results follow from this adaptive method. This section develops a weighted nonlinear project method (as opposed to Euclidean-type projection method) to sharpen the theoretical convergence rate and improve practical performance. The method is summarised in Algorithm 5.2 and its development is presented below.

5.2.1 Problem reparameterisation

The proposed method is an extension of the mirror descent (MD) algorithm introduced by Nemirovski and Yudin [76], and generalised in [48]. After successfully solving the computer tomography problem [8], mirror descent has attracted attention in the area of artificial intelligence, machine learning [47] and online optimisation [25]. Beck and Teboulle [4] show that MD can be viewed as a simple nonlinear subgradient projection, where the Bregman distance [24] is used in the projection operator instead of the usual Euclidean distance. In this thesis, a method is proposed based on the idea of nonlinear projection, where the log entropy distance function, a special class of Bregman distance, is utilised. Furthermore, since the feasible domain of the dual MRF consist of many disjoint sets, we use the average weighting parameters to combine the disjoint distances.

Firstly, recall the formulation of the dual LP-MRF:

$$\max_{\theta \in \Theta} \sum_{t \in T} E^t(\theta^t, x^t), \qquad \text{where} \quad \Theta = \left\{ \{\theta_i^t\} \middle| \sum_{t \in T} \theta_i^t = \hat{\theta}_i, \ \forall i \in \mathcal{I} \right\} \tag{5.18}$$

In the standard update (5.5), the subgradient is projected onto the entire feasible set Θ. The amount of change depends on a common step-size τ_k and so is the same at every node in disagreement. However, Θ is an intersection of $|\mathcal{I}|$ disjoint subsets. This motivates the specification of a subset-dependent projection for the disagreement nodes, reflecting the progress needed for each subset. The expectation is to derive a new projection scheme that should converge faster than the standard subgradient method. The new scheme allows the projection, on the corresponding subset, of every disagreement node to have a subset-dependent step-size. This can be achieved, with the weighted projection scheme, using the iterative updates (5.32) and subset-dependent step-sizes (5.48) employing the weighting parameters (5.42).

In addition, the key condition for any projection is to maintain the original unary potentials.

Consider the following reparameterisation that guarantees the feasibility of the unary potentials:

$$\max_{\rho\in\Delta,\lambda\in\Lambda} F(\rho,\lambda) := \max_{\rho\in\Delta,\lambda\in\Lambda} \sum_{t\in T} E^t(\rho^t.\hat{\theta}+\lambda^t, x^t) \tag{5.19}$$

The domains Δ and Λ are the union of the disjoint subsets:

$$\forall i \in \mathcal{I}: \quad \Delta :=_{\otimes} \Delta_i, \qquad \Lambda :=_{\otimes} \Lambda_i$$

where each subset is given by:

$$\Delta_i = \left\{ \{\rho_i^t\} \middle| \sum_{t\in T} \rho_i^t = 1;\ \rho_i^t \geq 0 \right\} \tag{5.20a}$$

$$\Lambda_i = \left\{ \{\lambda_i^t\} \middle| \sum_{t\in T} \lambda_i^t = 0 \right\} \tag{5.20b}$$

Without ρ, problem (5.19) is exactly the dual LP-MRF (5.18). The motivation for using ρ comes from the fact that, if the exact bound of the search space is known, then the entropic projection can be much more efficient than the Euclidean projection [8, 4]. In practice, most of the unary potentials were found to settle in the range $[0,\hat{\theta}]$. Furthermore, it will be shown in Theorem 5.2.9 that, in the worst case, the combination of entropic and Euclidean projection is better than Euclidean projection alone.

Lemma 5.2.1 *The reparameterised dual problem* (5.19) *is equivalent to the dual LP-MRF* (5.18)*, where:*

- *The problem* (5.19) *is a relaxation of* (5.18) *and the pair* (Δ,Λ) *preserve the original unary potentials.*

- *The optimal solutions* θ^* *of dual LP-MRF can be replaced by a pair of optimal* (ρ^*,λ^*) *that satisfies:*

$$\sum_{t\in T} E^t(\theta^{t(*)}) = F(\rho^*,\lambda^*) \geq \sum_{t\in T} E^t(\theta^t)$$

Proof • First, let us show that the sets Δ and Λ preserve the unary potentials. For any arbitrary pair ($\rho \in \Delta, \lambda \in \Lambda$), let $\theta_i^t = \rho_i^t.\hat{\theta}_i + \lambda_i^t$, $\forall i \in \mathcal{I}$ then:

$$\forall i \in \mathcal{I}: \quad \sum_{t \in T} \theta_i^t = \sum_{t \in T} \rho_i^t.\hat{\theta}_i + \lambda_i^t = \left(\sum_{t \in T} \rho_i^t\right).\hat{\theta}_i + \left(\sum_{t \in T} \lambda_i^t\right) = \hat{\theta}_i$$

Therefore, $\{\rho^t.\hat{\theta} + \lambda^t\} \in \Theta$. Problem (5.19) is a relaxation of (5.18) and is derived from the fact that we can define a one-to-many mapping between the set Θ to the pair (Δ, Λ), i.e. for any distribution $\{\theta^t\} \in \Theta$, an infinite number of distributions $\{\rho^t\}_k \in \Delta$ can be defined. For each $\{\rho^t\}_k \in \Delta$, one can compute a corresponding $\{\lambda^t\}_k$, such that:

$$\{\lambda^t\}_k = \{\theta^t\} - \{\rho^t\}_k.\hat{\theta} \Rightarrow \left\{\sum_{t \in T} \lambda^t\right\}_k = \sum_{t \in T} \theta^t - \left\{\sum_{t \in T} \rho^t.\hat{\theta}\right\}_k = 0 \Rightarrow \{\lambda^t\}_k \in \Lambda$$

Consequently, for any distribution $\{\theta^t\} \in \Theta$, there exist many pairs ρ_k, λ_k such that $\{\theta^t\} = \{\rho^t.\hat{\theta} + \lambda^t\}_k$ where $\{\rho^t\}_k \in \Delta$ and $\{\lambda^t\}_k \in \Lambda$.

- Assume that $(\tilde{\rho}, \tilde{\lambda})$ are the optimal solutions of (5.19), i.e.

$$F(\tilde{\rho}, \tilde{\lambda}) \geq F(\rho, \lambda), \ \forall(\rho \in \Delta, \lambda \in \Lambda)$$

Let:

$$\tilde{\theta}^t = \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t \Rightarrow \sum_{t \in T} \tilde{\theta}^t = \hat{\theta} \Rightarrow \{\tilde{\theta}^t\} \in \Theta$$

Since θ^* is the optimal solution of dual LP-MRF, we have:

$$\sum_{t \in T} E^t(\theta^{t(*)}) \geq \sum_{t \in T} E^t(\tilde{\theta}^t) = F(\tilde{\rho}, \tilde{\lambda})$$

However, as shown above, it is always possible to find at least one pair ($\rho^* \in \Delta, \lambda^* \in \Lambda$) such that:

$$\{\theta^{t(*)}\} = \{\rho^{t(*)}.\hat{\theta} + \lambda^{t(*)}\}$$

Using the assumption that $(\tilde{\rho}, \tilde{\lambda})$ are the optimal solutions of (5.19), gives:

$$\sum_{t \in T} E^t(\theta^{t(*)}) \geq \sum_{t \in T} E^t(\tilde{\theta}^t) = F(\tilde{\rho}, \tilde{\lambda}) \geq F(\rho^*, \lambda^*) = \sum_{t \in T} E^t(\theta^{t(*)})$$

which requires all inequalities to be trictly equal.

■

5.2.2 Mirror Descent (MD) setup

It has been shown in [4] that MD is a generalisation of the projected subgradient method, where the projection is performed on some nonlinear distance functions. One can write the MD updates as a sequence of the proximal algorithm:

$$\begin{bmatrix}\rho^{(k+1)}\\ \lambda^{(k+1)}\end{bmatrix} = \underset{\rho\in\Delta,\lambda\in\Lambda}{\arg\max}\left\{\begin{array}{l}\quad\left\langle F'(\rho^{(k)}),\rho\right\rangle - \frac{1}{\tau_k}D_\Delta(\rho,\rho^{(k)})\\ + \\ \quad\left\langle F'(\lambda^{(k)}),\lambda\right\rangle - \frac{1}{\eta_k}D_\Lambda(\lambda,\lambda^{(k)})\end{array}\right\} \tag{5.21}$$

Since the function F is linear in both variables, and since ρ and λ are decoupled, the *subgradients* with respect to each vector ρ, λ are disjoint. The *weighted distances* D_Δ, D_Λ and step-sizes τ, η are defined independently to exploit the geometry of each set. Each distance function will be equipped with a compatible pair of *norm* and *dual norm.* Based on the definitions of specialised norms, one can derive subset-dependent *Lipschitz* and *weighted Lipschitz.* The key ingredients required for the MD method are described next.

Subgradient. A subgradient is chosen such that the computational cost per iteration is as low as the Euclidean projected subgradient method (5.5). Since $F(\rho, \lambda)$ is linear and the vectors (ρ, λ) are disjoint, the subgradient consists of two independent parts:

Lemma 5.2.2 *Let us define a vector:*

$$F'(\rho,\lambda) \overset{\text{def}}{=} [\hat{\theta}.\bar{x}\ ;\ \bar{x}] = [\hat{\theta}.\{\bar{x}^t\}\ ;\ \{\bar{x}^t\}] \tag{5.22a}$$

$$\text{where} \qquad \bar{x}^t = \arg\min_{x^t\in X^t}\langle \rho^t.\hat{\theta} + \lambda^t, x^t\rangle \tag{5.22b}$$

Then $F'(\rho, \lambda)$ is a subgradient of $F(\rho, \lambda)$, i.e. it satisfies the subgradient inequality of concave function:

$$\forall(\tilde{\rho}\in\Delta_{-\rho}, \tilde{\lambda}\in\Lambda_{-\lambda}): \quad F(\tilde{\rho},\tilde{\lambda}) \leq F(\rho,\lambda) + \langle\hat{\theta}.\bar{x}, \tilde{\rho}-\rho\rangle + \langle\bar{x}, \tilde{\lambda}-\lambda\rangle$$

Proof By the definition (5.22b), $\bar{x}^t$ is not optimal for $\min\limits_{x^t \in X^t} \langle \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t, x^t \rangle$, i.e.

$$\forall t \in T: \quad \min_{x^t \in X^t} \langle \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t, x^t \rangle \leq \langle \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t, \bar{x}^t \rangle$$

In addition,

$$\begin{aligned}
F(\tilde{\rho}, \tilde{\lambda}) &= \sum_{t \in T} \min_{x^t \in X^t} \langle \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t, x^t \rangle \leq \sum_{t \in T} \langle \tilde{\rho}^t.\hat{\theta} + \tilde{\lambda}^t, \bar{x}^t \rangle \\
F(\tilde{\rho}, \tilde{\lambda}) &\leq \sum_{t \in T} \Big[\langle \rho^t.\hat{\theta} + \lambda^t, \bar{x}^t \rangle + \langle \hat{\theta}.\bar{x}^t, \tilde{\rho}^t - \rho^t \rangle + \langle \bar{x}^t, \tilde{\lambda}^t - \lambda^t \rangle \Big] \\
&= F(\rho, \lambda) + \langle \hat{\theta}.\bar{x}, \tilde{\rho} - \rho \rangle + \langle \bar{x}, \tilde{\lambda} - \lambda \rangle
\end{aligned}$$

■

Subset distance functions and subset norms. MD updates apply subgradient projection on some nonlinear distance equipped with a *compatible* norm. The domain Δ and Λ have the form of the direct product of non-overlapping subsets:

$$\Delta = \Delta_1 \times \Delta_2 \times ... \times \Delta_{|\mathcal{I}|} \qquad \Lambda = \Lambda_1 \times \Lambda_2 \times ... \times \Lambda_{|\mathcal{I}|}$$

Instead of defining one distance measure over the whole domain, giving the same domain as in the standard projection method, a combination of weighted subset distances is used. The nonlinear distance functions associated with each subset are defined by:

$$\begin{aligned}
D^i_\Delta(u_i, v_i) &= \psi^i_\Delta(u_i) - \psi^i_\Delta(v_i) - \langle \nabla \psi^i_\Delta(v_i), u_i - v_i \rangle \\
D^i_\Lambda(u_i, v_i) &= \psi^i_\Lambda(u_i) - \psi^i_\Lambda(v_i) - \langle \nabla \psi^i_\Lambda(v_i), u_i - v_i \rangle
\end{aligned}$$

The above definitions are based on the Bregman distance in proximal algorithms [51], where ψ is a distance generating function (d.g.f) that is required to be 1-strongly convex with respect to a compatible norm.

Lemma 5.2.3 *For each subset, a distance function and its compatible norm are defined as:*

- *For the simplex set Δ_i, let:*

$$\psi_\Delta^i(\rho_i) = \sum_{t\in T} \rho_i^t \log \rho_i^t, \text{ if } \rho_i \in \Delta_i;\ else, +\infty \tag{5.23a}$$

In addition, adopting the convention $0 \log 0 \equiv 0$, ψ_Δ^i is 1-strongly convex with respect to the l_1-norm $\|.\|_1$

- *For the linear constraint Λ_i, let:*

$$\psi_\Lambda^i(\lambda_i) = \frac{1}{2}\sum_{t\in T} (\lambda_i^t)^2, \text{ if } \lambda_i \in \Lambda_i;\ else, +\infty \tag{5.23b}$$

then ψ_Λ^i is 1-strongly convex with respect to the l_2-norm $\|.\|_2$

The 1-strongly convex proof for the log entropy d.g.f ψ_Δ^i can be found in [4, Proposition 5.1]. The proof for the latter statement, strongly convex of (5.23b), can be directly derived from the strong convexity inequalities of the gradient of ψ_Λ^i. With the d.g.f defined above, each subset Δ_i is equipped with a log entropy distance D_Δ^i and $\|.\|_1$; while the pair $(D_\Lambda^i, \|.\|_2)$ are just Euclidean settings for each subset Λ_i.

Weighted distances and weighted norms. For each set Δ and Λ, consider a weighted average distance associated of the form:

$$\begin{aligned} D_\Delta(u,v) &= \sum_{i\in\mathcal{I}} \alpha_\Delta^i D_\Delta^i(u_i,v_i) = \sum_{i\in\mathcal{I}} \alpha_\Delta^i \left[\psi_\Delta^i(u_i) - \psi_\Delta^i(v_i) - \langle \nabla\psi_\Delta^i(v_i), u_i - v_i\rangle\right] \\ &\overset{\text{def}}{=} \psi_\Delta(u) - \psi_\Delta(v) - \langle \nabla\psi_\Delta(v), u - v\rangle \qquad (5.24a) \\ D_\Lambda(u,v) &= \sum_{i\in\mathcal{I}} \alpha_\Lambda^i D_\Lambda^i(u_i,v_i) = \sum_{i\in\mathcal{I}} \alpha_\Lambda^i \left[\psi_\Lambda^i(u_i) - \psi_\Lambda^i(v_i) - \langle \nabla\psi_\Lambda^i(v_i), u_i - v_i\rangle\right] \\ &\overset{\text{def}}{=} \psi_\Lambda(u) - \psi_\Lambda(v) - \langle \nabla\psi_\Lambda(v), u - v\rangle \qquad (5.24b) \end{aligned}$$

where $\alpha_\Delta^i, \alpha_\Lambda^i > 0$ are the weighting parameters which will be optimised in (5.42). The above weighted distances have the form of Bregman distance with weighted d.g.f:

$$\psi_\Delta(\rho) = \sum_{i\in\mathcal{I}} \alpha_\Delta^i \psi_\Delta^i(\rho_i) \quad \text{and} \quad \psi_\Lambda(\lambda) = \sum_{i\in\mathcal{I}} \alpha_\Lambda^i \psi_\Lambda^i(\lambda_i) \tag{5.25}$$

Each weighted distance also requires a compatible norm, where its weighted d.g.f is 1-strongly convex with the defined norm. To this end, consider the weighted norms associated with each weighted d.g.f:

$$\|\rho\|_\Delta = \sqrt{\sum_{i\in\mathcal{I}} \alpha^i_\Delta \|\rho_i\|_1^2} \quad \text{and} \quad \|\lambda\|_\Lambda = \sqrt{\sum_{i\in\mathcal{I}} \alpha^i_\Lambda \|\lambda_i\|_2^2} \tag{5.26}$$

Lemma 5.2.4 *Strong Convexity of weighted d.g.f:*

- *Let* $\psi_\Delta : \Delta \to \mathbb{R}$ *be the weighted d.g.f defined in* (5.25)*, then* ψ_Δ *is* 1*-strongly convex w.r.t the weighted norm* $\|.\|_\Delta$*.*
- *Let* $\psi_\Lambda : \Lambda \to \mathbb{R}$ *be the weighted d.g.f defined in* (5.25)*, then* ψ_Λ *is* 1*-strongly convex w.r.t the weighted norm* $\|.\|_\Lambda$*.*

Proof The proof works in the same way for both cases, so we omit the suffix Δ and Λ in all symbols.

$$\begin{aligned} \langle \nabla\psi(u) - \nabla\psi(v), u - v\rangle &= \sum_{i\in\mathcal{I}} \alpha^i \langle \nabla\psi^i(u_i) - \nabla\psi^i(v_i), u_i - v_i\rangle \\ &\geq \sum_{i\in I} \alpha^i \|u_i - v_i\|^2 \quad (\text{either } \|.\|_1 \text{ or } \|.\|_2) \qquad (5.27) \\ &= \|u - v\|^2 \quad (\text{either } \|.\|_\Delta \text{ or } \|.\|_\Lambda) \end{aligned}$$

The inequality (5.27) comes from Lemma 5.2.3. ∎

Dual Norm, local Lipschitz and weighted Lipschitz. For each weighted norm, a corresponding conjugate norm can be derived [14]:

$$\|\rho\|_{\Delta*} = \sqrt{\sum_{i\in\mathcal{I}} \|\rho_i\|_\infty^2 / \alpha^i_\Delta} \quad \text{and} \quad \|\lambda\|_{\Lambda*} = \sqrt{\sum_{i\in\mathcal{I}} \|\lambda_i\|_2^2 / \alpha^i_\Lambda} \tag{5.28}$$

The definitions of dual norms suggest the forms of the weighted Lipschitz constants as a combination of the local Lipschitz constants associated with every disjoint subset. The local Lipschitz constants can be easily computed given the input data:

$$\mathcal{L}_{\Delta_i} = \sup_{\rho_i\in\Delta_i} \|F'(\rho_i)\|_\infty = |\hat{\theta}_i| \tag{5.29a}$$

$$\mathcal{L}_{\Lambda_i} = \sup_{\lambda_i\in\Lambda_i} \|F'(\lambda_i)\|_2 = \sqrt{|T|} \tag{5.29b}$$

The subgradient $F'(\rho_i^t) = \hat{\theta}_i.\bar{x}_i^t \in \{\hat{\theta}_i, 0\}$, thus giving (5.29a). The subgradient $F'(\lambda_i^t) = \bar{x}_i^t$ admits binary values $\{1, 0\}$. The worst case, of every tree being assigned the same label, results in (5.29b). With the definitions of local subset-dependent Lipschitz numbers, the average Lipschitz constants are then given by:

$$\begin{aligned}\mathcal{L}_\Delta &= \sup_{\rho\in\Delta} \|F'(\rho)\|_{\Delta*} = \sqrt{\sum_{i\in\mathcal{I}} \mathcal{L}^2_{\Delta_i}/\alpha^i_\Delta} &\quad (5.30a)\\ \mathcal{L}_\Lambda &= \sup_{\lambda\in\Lambda} \|F'(\lambda)\|_{\Lambda*} = \sqrt{\sum_{i\in\mathcal{I}} \mathcal{L}^2_{\Lambda_i}/\alpha^i_\Lambda} &\quad (5.30b)\end{aligned}$$

The analytical forms of the average Lipschitz are surprisingly simple once the optimal weighting parameters are derived. At this point, there are sufficient ingredients to set up the MD method and examine its convergence properties.

5.2.3 Proximal updates

A subset-dependent step-size projection scheme was motivated in Section 5.2.1. As will be described, this scheme is derived using the proximal updates with the weighted distance function (5.24). The *subset-dependent* step-size arises as the weighting parameters α^i are combined with the common step-size τ. The proximal update (5.21) is linear in both ρ and λ, therefore $\rho^{(k+1)}$ and $\lambda^{(k+1)}$ can be computed independently by:

$$\begin{aligned}\rho^{(k+1)} &= \arg\max_{\rho\in\Delta} \left\{ \langle \rho, F'(\rho^{(k)})\rangle - \frac{1}{\tau_k} D_\Delta(\rho, \rho^{(k)}) \right\}\\ &= \arg\max_{\rho\in\Delta} \left\{ \tau_k \langle \rho, F'(\rho^{(k)})\rangle - \psi_\Delta(\rho) + \langle \nabla\psi_\Delta(\rho^{(k)}), \rho\rangle \right\} &\quad (5.31a)\\ \lambda^{(k+1)} &= \arg\max_{\lambda\in\Lambda} \left\{ \langle \lambda, F'(\lambda^{(k)})\rangle - \frac{1}{\eta_k} D_\Lambda(\lambda, \lambda^{(k)}) \right\}\\ &= \arg\max_{\lambda\in\Lambda} \left\{ \eta_k \langle \rho, F'(\lambda^{(k)})\rangle - \psi_\Lambda(\lambda) + \langle \nabla\psi_\Lambda(\lambda^{(k)}), \lambda\rangle \right\} &\quad (5.31b)\end{aligned}$$

Proposition 5.2.5 *$\forall i \in \mathcal{I}$, the solutions of proximal sequences are given explicitly by:*

$$\begin{aligned}\rho_i^{t(k+1)} &= \frac{\rho_i^{t(k)} \exp\left(F'(\rho_i^{t(k)}).\tau_k/\alpha^i_\Delta\right)}{\sum_{t\in T} \rho_i^{t(k)} \exp\left(F'(\rho_i^{t(k)}).\tau_k/\alpha^i_\Delta\right)} &\quad (5.32a)\\ \lambda_i^{t(k+1)} &= \frac{\eta_k}{\alpha^i_\Lambda}\left(F'(\lambda_i^{t(k)}) - \frac{\sum_{t\in T} F'(\lambda_i^{t(k)}}{|T|} \right) &\quad (5.32b)\end{aligned}$$

Proof In this proof, updates for ρ are shown, with λ derived similarly. The Lagrangian of (5.31a) is given by:

$$\begin{aligned}
L(\rho,\gamma) &= \sum_{i\in\mathcal{I}}\left[\sum_{t\in T}\left(\tau_k.\rho_i^t.F'(\rho_i^{t(k)}) - \alpha_\Delta^i.\left(\rho_i^t\log\rho_i^t - \nabla\psi_\Delta^i(\rho_i^{t(k)}).\rho_i^t\right)\right) + \gamma_i\left(\sum_{t\in T}\rho_i^t - 1\right)\right]\\
\nabla_{\rho_i^t}L &= \tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i - \left(\log\rho_i^t - \log\rho_i^{t(k)}\right) + \gamma_i/\alpha_\Delta^i = 0\\
\Rightarrow \log\rho_i^t &= \log\rho_i^{t(k)} + \gamma_i/\alpha_\Delta^i + \tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\\
\Rightarrow \rho_i^t &= \left[\rho_i^{t(k)}\exp\left(\tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\right)\right]\exp\left(\gamma_i/\alpha_\Delta^i\right)\\
\nabla_{\gamma_i}L &= \sum_{t\in T}\rho_i^t - 1 = 0\\
\Rightarrow 1 &= \exp\left(\gamma_i/\alpha_\Delta^i\right)\sum_{t\in T}\left[\rho_i^{t(k)}\exp\left(\tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\right)\right]\\
\Rightarrow \exp\left(\gamma_i/\alpha_\Delta^i\right) &= \left[\sum_{t\in T}\rho_i^{t(k)}\exp\left(\tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\right)\right]^{-1}\\
\Rightarrow \rho_i^t &= \frac{\rho_i^{t(k)}\exp\left(\tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\right)}{\sum_{t\in T}\rho_i^{t(k)}\exp\left(\tau_k F'(\rho_i^{t(k)})/\alpha_\Delta^i\right)}
\end{aligned}$$

Applying the same technique leads to the update for λ_i^t in the form (5.32b). ■

5.2.4 Convergence analysis

The MD iterations (5.32) solve for ρ and λ independently. Since the two variables are disjoint, MD can either update them simultaneously or sequentially. By examining the convergence properties a theoretical justification is provided, showing that sequential updates lead to a faster convergence rate, i.e. MD updates ρ first, until there is no improvement in the dual, then it switches to update λ. In addition, via the optimality bound, the optimal step-size for the weighted entropic projection on the set Δ can be computed, as well as an adaptive step-size strategy for weighted Euclidean projection.

Theorem 5.2.6 *The MD method provides a better optimality bound for sequential updates than for parallel updates. Let* $F^* \overset{\text{def}}{=} F(\rho^*,\lambda^*)$ *denote the optimal objective value, and let* $\bar{F}_i^j = \max_{k=i,..,j} F_k$*, then the optimal bound is given by:*

$$F^* - \bar{F}_1^K \leq \frac{\mathcal{L}_\Delta\sqrt{2\Omega_\Delta}\sqrt{k_1} + \mathcal{L}_\Lambda\sqrt{2\Omega_\Lambda}\sqrt{k_2}}{K} \tag{5.33}$$

where $\Omega_\Delta = \sup D_\Delta(\rho^*, \rho^1)$ *and* $\Omega_\Lambda = \sup D_\Lambda(\lambda^*, \lambda^{k_1+1})$*, with* ρ^1, λ^{k_1+1} *being the starting points of the entropic and Euclidean projections respectively;* k_1 *is the number of entropy proximal updates,* k_2 *is the iterations count for Euclidean proximal updates; and* $K = k_1 + k_2$ *is the total number of iterations.*

Proof As the sequences ρ and λ can be computed independently by equations (5.31), the convergence analysis is similar for both cases apart from the differences in the distances, norms and Lipschitz constants. Adapting the proof of [48, Proposition 1.1], we produce the convergence estimate for the first sequence; the derivation of the latter sequence follows straightforwardly. The optimality condition of (5.31a) is given by:

$$\begin{aligned} 0 &\leq \langle \rho^* - \rho^{k+1}, -\tau_k F'(\rho^k) + \nabla\psi_\Delta^{k+1} - \nabla\psi_\Delta^k \rangle \\ \tau_k\langle F'(\rho^k), \rho^* - \rho^k\rangle &\leq \langle \nabla\psi_\Delta^{k+1} - \nabla\psi_\Delta^k, \rho^* - \rho^{k+1}\rangle + \tau_k\langle F'(\rho^k), \rho^{k+1} - \rho^k\rangle \\ &= D_\Delta(\rho^*, \rho^k) - D_\Delta(\rho^*, \rho^{k+1}) + \left[\underbrace{-D_\Delta(\rho^{k+1}, \rho^k) + \tau_k\langle F'(\rho^k), \rho^{k+1} - \rho^k\rangle}_{\delta}\right] \end{aligned}$$

From lemma 5.2.4, ψ_Δ is 1-strongly convex w.r.t $\|.\|_\Delta$, i.e.

$$\psi_\Delta(\rho^{k+1}) \geq \psi_\Delta(\rho^k) + \langle \nabla\psi_\Delta^k, \rho^{k+1} - \rho^k\rangle + \frac{1}{2}\|\rho^{k+1} - \rho^k\|_\Delta^2$$

It follows that:

$$\begin{aligned} \delta &\leq \tau_k\langle F'(\rho^k), \rho^{k+1} - \rho^k\rangle - \frac{1}{2}\|\rho^{k+1} - \rho^k\|_\Delta^2 \\ &\leq \tau_k\|F'(\rho^k)\|_{\Delta*}.\|\rho^{k+1} - \rho^k\|_\Delta - \frac{1}{2}\|\rho^{k+1} - \rho^k\|_\Delta^2 && (5.34a) \\ &\leq \max_s\left\{\tau_k\|F'(\rho^k)\|_{\Delta*}.s - \frac{1}{2}.s^2\right\} && (5.34b) \\ &= \frac{\tau_k^2\|F'(\rho^k)\|_{\Delta*}^2}{2} && (5.34c) \end{aligned}$$

Replacing $\|\rho^{k+1} - \rho^k\|_\Delta$ in equation (5.34b) by s, and maximising over s, gives (5.34a) $\leq$ (5.34b). The maximum (5.34c) is obtained at $s = \tau_k\|F'(\rho^k)\|_{\Delta*}$. In addition, the subgradient inequality of the concave function $F(\rho)$ gives:

$$\begin{aligned} \tau_k\left(F(\rho^*) - F(\rho^k)\right) &\leq \tau_k\langle F'(\rho^k), \rho^* - \rho^k\rangle \\ \tau_k(F^* - F^k) &\leq D_\Delta(\rho^*, \rho^k) - D_\Delta(\rho^*, \rho^{k+1}) + \frac{\tau_k^2\|F'(\rho^k)\|_{\Delta*}^2}{2} && (5.35) \end{aligned}$$

Summing up the inequality (5.35) over $k = 1, ..., k_1$ and taking into account the inequalities $D_\Delta(\rho^*, \rho^{k_1+1}) \geq 0$ and $\bar{F}_1^{k_1} \geq F(\rho^k)$, yields:

$$\begin{aligned}
\sum_{k=1}^{k_1} \tau_k \left(F^* - \bar{F}_1^{k_1}\right) &\leq D_\Delta(\rho^*, \rho^1) + \frac{1}{2}\sum_{k=1}^{k_1} \tau_k^2 \|F'(\rho^k)\|_{\Delta*}^2 \\
F^* - \bar{F}_1^{k_1} &\leq \frac{2\Omega_\Delta + \mathcal{L}_\Delta^2 \left(\sum_{k=1}^{k_1} \tau_k^2\right)}{2\sum_{k=1}^{k_1} \tau_k} \qquad (5.36)
\end{aligned}$$

where $\mathcal{L}_\Delta$ is the average Lipschitz constant (5.30a). Minimising the RHS of (5.36) w.r.t. $\{\tau_k\}$, leads to the optimal bound:

$$F^* - \bar{F}_1^{k_1} \leq \frac{\sqrt{2\Omega_\Delta}\mathcal{L}_\Delta}{\sqrt{k_1}}$$

with the constant step-size:

$$\tau = \frac{\sqrt{2\Omega_\Delta}}{\mathcal{L}_\Delta \sqrt{k_1}} \qquad (5.37a)$$

Applying the same technique, leads to the following optimal constant step-size for the weighted Euclidean:

$$\eta = \frac{\sqrt{2\Omega_\Lambda}}{\mathcal{L}_\Lambda \sqrt{k_2}} \qquad (5.37b)$$

Using the optimal constant step-sizes, the inequalities (5.35) can be shown to be valid for both types of projected iteration:

$$\begin{aligned}
F^* - F^k &\leq \frac{D_\Delta(\rho^*, \rho^k)}{\tau} - \frac{D_\Delta(\rho^*, \rho^{k+1})}{\tau} + \frac{\tau\|F'(\rho^k)\|_{\Delta*}^2}{2} \quad \text{for } k = 1, ..., k_1 \qquad (5.38a) \\
F^* - F^k &\leq \frac{D_\Lambda(\lambda^*, \lambda^k)}{\lambda} - \frac{D_\Lambda(\lambda^*, \lambda^{k+1})}{\lambda} + \frac{\lambda\|F'(\lambda^k)\|_{\Lambda*}^2}{2} \quad \text{for } k = k_1 + 1, ..., K \qquad (5.38b)
\end{aligned}$$

Summing up (5.38) over K iterations gives:

$$\begin{aligned}
K(F^* - \bar{F}_1^K) &\leq \sum_{k=1}^{K}(F^* - F_k) \leq \frac{D_\Delta(\rho^*, \rho^1)}{\tau} + \frac{D_\Lambda(\lambda^*, \lambda^{k_1+1})}{\eta} + \frac{k_1\tau\mathcal{L}_\Delta^2 + k_2\eta^2\mathcal{L}_\Lambda^2}{2} \\
F^* - \bar{F}_1^K &\leq \frac{\Omega_\Delta}{K\tau} + \frac{\Omega_\Lambda}{K\eta} + \frac{1}{2}\left(\frac{k_1}{K}\tau\mathcal{L}_\Delta^2 + \frac{k_2}{K}\eta\mathcal{L}_\Lambda^2\right) \qquad (5.39)
\end{aligned}$$

If the updates are done in parallel for both ρ and λ at every iteration, then the convergence analysis is similar with the iteration count $K = k_1 = k_2$. In such a case,

$$\underbrace{(5.39)}_{\text{sequential bound}} \quad < \quad \underbrace{\frac{\Omega_\Delta}{K\tau} + \frac{\Omega_\Lambda}{K\eta} + \frac{1}{2}(\tau\mathcal{L}_\Delta^2 + \eta\mathcal{L}_\Lambda^2)}_{\text{parallel bound}}$$

The above inequality shows that sequential updates provide a better optimality bound compared to parallel updates. Substituting the optimal step-size strategies (5.37) into Equation (5.39) results in (5.33):

$$F^* - \bar{F}_1^K \leq \frac{\mathcal{L}_\Delta\sqrt{2\Omega_\Delta}\sqrt{k_1} + \mathcal{L}_\Lambda\sqrt{2\Omega_\Lambda}\sqrt{k_2}}{K}$$

∎

The unknown maximum weighted distances $\Omega_\Delta, \Omega_\Lambda$ can be written as the weighted combinations of the maximum distance of each disjoint subset:

$$\Omega_\Delta = \sum_{i\in\mathcal{I}} \alpha_\Delta^i \Omega_\Delta^i \quad \text{and} \quad \Omega_\Lambda = \sum_{i\in\mathcal{I}} \alpha_\Lambda^i \Omega_\Lambda^i$$

where each distance quantity $i \in \mathcal{I}$ denotes the maximum distance from the starting point to the optimal point, i.e.

$$\Omega_\Delta^i = \sup D_\Delta^i(\rho_i^*, \rho_i^{(1)}) \quad \text{and} \quad \Omega_\Lambda^i = \sup D_\Lambda^i(\lambda_i^*, \lambda_i^{(k_1+1)})$$

Whilst the quantity

$$\Omega_\Lambda^i = \sup \frac{1}{2}\|\lambda_i^* - \lambda_i^{(k_1+1)}\|_2^2 \tag{5.40a}$$

can only be estimated based on the primal-dual gap, the maximum simplex subset can be computed analytically.

Lemma 5.2.7 *For the choice of $\rho_i^{t(1)} = \frac{1}{|T|}$, the following simplex bound exists on the subset Δ_i:*

$$\Omega_\Delta^i = \log|T| \tag{5.40b}$$

Proof Using the d.g.f definition (5.23a) for all $\rho_i^t \in \Delta_i$:

$$D_\Delta^i(\rho_i^*, \rho_i^{(1)}) = \sum_{t\in T} \rho_i^{t(*)} \log \frac{\rho_i^{t(*)}}{\rho_i^{t(1)}} = \sum_{t\in T} \rho_i^{t(*)} \log \rho_i^{t(*)} + \left(\sum_{t\in T} \rho_i^{t(*)}\right) \log|T| \leq \log|T|$$

■

At this point, the information about the local Lipschitz (5.29) and the maximum subset distance (5.40) is known, the remaining unknown quantities in the optimality bound are the weighting parameters α^i_Δ and α^i_Λ. The optimal values of these weighting parameters are those which minimise the RHS of (5.33).

Lemma 5.2.8 *Let an arbitrary weighted Lipschitz constant be defined as* $\mathcal{L} = \sqrt{\sum_{i\in\mathcal{I}} \mathcal{L}_i^2/\alpha^i}$ *and an arbitrary weighted distance defined by* $\Omega = \sum_{i\in\mathcal{I}} \alpha^i \Omega_i$*. The quantity* $\mathcal{L}\sqrt{\Omega}$ *is then minimised by:*

$$\min_{\{\alpha^i\}} \mathcal{L}\sqrt{\Omega} = \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega_i} \tag{5.41}$$

at

$$\forall i \in \mathcal{I}: \quad \alpha^i = \frac{\mathcal{L}_i}{\sqrt{\Omega^i}\left[\sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega^i}\right]} \tag{5.42}$$

Furthermore, at the optimal $\{\alpha^i\}$*, we have:*

$$\Omega = 1 \tag{5.43}$$

and

$$\mathcal{L} = \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega^i} \tag{5.44}$$

Proof Since all quantities are positive, it can be seen that:

$$\arg\min_{\{\alpha^i\}} \mathcal{L}\sqrt{\Omega} = \arg\min_{\{\alpha^i\}} \mathcal{L}^2\Omega$$

Minimising $\mathcal{L}^2\Omega$, we obtain:

$$\forall i \in \mathcal{I}: \quad \alpha^i = \frac{\mathcal{L}_i\sqrt{\Omega}}{\mathcal{L}\sqrt{\Omega_i}} \tag{5.45a}$$

Thus,

$$\begin{aligned}\Omega &= \sum_{i\in\mathcal{I}} \alpha^i \Omega_i = \frac{\sqrt{\Omega}}{\mathcal{L}} \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega_i} \\ \Rightarrow \sqrt{\Omega} &= \frac{1}{\mathcal{L}} \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega_i} \qquad (5.45\text{b}) \\ \Rightarrow \mathcal{L} &= \frac{1}{\sqrt{\Omega}} \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega_i} \qquad (5.45\text{c})\end{aligned}$$

Substituting (5.45b) and (5.45c) into (5.45a), gives:

$$\alpha^i = \frac{\mathcal{L}_i}{\sqrt{\Omega^i}\left[\sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega^i}\right]} \ , \quad \Omega = 1 \ , \quad \mathcal{L} = \sum_{i\in\mathcal{I}} \mathcal{L}_i \sqrt{\Omega^i}$$

then (5.41) is followed. ∎

The next theorem states the explicit optimality bound and re-establishes the fact that sequential updates are faster than parallel updates. In addition, it will be shown that the Mirror Descent method with weighted projection provides a much lower optimality bound compared to the standard projected subgradient method.

Theorem 5.2.9 *$\forall i \in \mathcal{I}$, let $\rho_i^{t(1)} = \frac{1}{|T|}, \lambda_i^{t(k_1+1)} = 0$ then we have the following optimality bound:*

$$F^* - \bar{F} \leq \frac{\sqrt{k_1}}{K}\sqrt{2}\sum_{i\in\mathcal{I}} |\hat{\theta}_i| \sqrt{\log|T|} + \frac{\sqrt{k_2}}{K} \sum_{i\in\mathcal{I}} \|\lambda_i^*\|_2 \sqrt{|T|} \qquad (5.46)$$

where $\bar{F} = \max_{k=i,..,K} F_k$. Furthermore, this bound is smaller than:

1. *Parallel Mirror Descent updates.*
2. *Weighted Euclidean Projection updates (without entropic projection).*
3. *Standard Euclidean Projection.*

Proof From the Theorem 5.2.6 and Lemma 5.2.8, we have:

$$F^* - \bar{F} \leq \frac{\sqrt{k_1}}{K}\sqrt{2}\sum_{i\in\mathcal{I}} \mathcal{L}_{\Delta_i} \sqrt{\Omega^i_\Delta} + \frac{\sqrt{k_2}}{K}\sqrt{2}\sum_{i\in\mathcal{I}} \mathcal{L}_{\Lambda_i} \sqrt{\Omega^i_\Lambda} \qquad (5.47\text{a})$$

1. If parallel updates were used, then $k_1 = K$ and $k_2 = K$, resulting in:

$$(5.47\text{a}) < \frac{1}{\sqrt{K}}\sqrt{2}\sum_{i\in\mathcal{I}}\mathcal{L}_{\Delta_i}\sqrt{\Omega^i_\Delta} + \frac{1}{\sqrt{K}}\sqrt{2}\sum_{i\in\mathcal{I}}\mathcal{L}_{\Lambda_i}\sqrt{\Omega^i_\Lambda}$$

   Substituting the definitions of maximum distance (5.40) and Lipschitz (5.29) of individual subsets into (5.47a) leads to the optimal bound (5.46):

$$F^* - \bar{F} \leq \frac{\sqrt{k_1}}{K}\sqrt{2}\sum_{i\in\mathcal{I}}|\hat{\theta}_i|\sqrt{\log|T|} + \frac{\sqrt{k_2}}{K}\sum_{i\in\mathcal{I}}\|\lambda^*_i\|_2\sqrt{|T|}$$

2. If only weighted Euclidean projection was used, the optimality bound has the form:

$$F^* - \bar{F} \leq \frac{1}{\sqrt{K}}\sum_{i\in\mathcal{I}}\|\hat{\theta}_i + \lambda^*_i\|_2\sqrt{|T|} \tag{5.47b}$$

   In the worst case, $\theta^{t(*)}_i = \hat{\theta}_i + \lambda^{t(*)}_i \notin [0, \hat{\theta}_i]$, i.e. λ^t_i needs to search over the full domain $[0, \hat{\theta}_i]$ plus the additional domain $[\hat{\theta}_i, \hat{\theta}_i + \lambda^{t(*)}_i]$, giving:

$$(5.47\text{b}) \equiv \frac{1}{\sqrt{K}}\sum_{i\in\mathcal{I}}|T||\hat{\theta}_i| + \sqrt{|T|}\|\lambda^*_i\|_2 > \frac{\sqrt{k_1}}{K}\sqrt{2}\sum_{i\in\mathcal{I}}|\hat{\theta}_i|\sqrt{\log|T|} + \frac{\sqrt{k_2}}{K}\sum_{i\in\mathcal{I}}\|\lambda^*_i\|_2\sqrt{|T|}$$

3. In the worst case, the convergence rate of the standard Euclidean projection (5.11) is given by:

$$F(\theta^*) - F(\bar{\theta}) \leq \frac{\sqrt{\left(\sum_{i\in\mathcal{I}}\sup\|\theta^{(1)}_i - \theta^*_i\|^2_2\right)|\mathcal{I}||T|}}{\sqrt{K}} = \frac{\sqrt{|T|}}{\sqrt{K}}\sqrt{\left(\sum_{i\in\mathcal{I}}\|\hat{\theta}_i + \lambda^*_i\|^2_2\right)|\mathcal{I}|} \tag{5.47c}$$

   This bound is greater than (5.47b) due to the Cauchy-Schwarz inequality:

$$\left(\sum_{i\in\mathcal{I}}\|\hat{\theta}_i + \lambda^*_i\|^2_2\right)|\mathcal{I}| \geq \left(\sum_{i\in\mathcal{I}}\|\hat{\theta}_i + \lambda^*_i\|_2\right)^2$$

   Therefore, in the worst case, the weighted MD method provides the fastest convergence rate:

$$(5.46) < (5.47\text{b}) \leq (5.47\text{c})$$

$\blacksquare$

5.2.5 Implementation

In addition to the improved convergence rate, the MD method also preserves the efficient implementation properties of the standard Euclidean projection method (Corollary 5.1.4 and 5.1.5). As before, it is only necessary to allocate the memory to store the tree collection $\{\theta^t\}, \forall t \in T$, and the updates for the MD sequence are only required at a few disagreeing nodes and labels. This section demonstrates this fact by deriving the explicit MD updates as stated in Proposition 5.2.5. The remaining quantities needed to define in the updates (5.32a),(5.32b) are τ_k/α^i_Δ and η_k/α^i_Λ. Using Equations (5.29), (5.37), (5.40), (5.42), (5.43) and (5.44), we have:

$$\frac{\tau_k}{\alpha^i_\Delta} = \frac{\sqrt{2\Omega_\Delta}}{\alpha^i_\Delta \mathcal{L}_\Delta \sqrt{k}} = \frac{\sqrt{2\Omega^i_\Delta}\mathcal{L}_\Delta}{\mathcal{L}_{\Delta_i}\mathcal{L}_\Delta \sqrt{k}} = \frac{\sqrt{2\log|T|}}{|\hat{\theta}_i|\sqrt{k}} \tag{5.48a}$$

$$\frac{\eta_k}{\alpha^i_\Lambda} = \frac{\sqrt{2\Omega_\Lambda}}{\alpha^i_\Lambda \mathcal{L}_\Lambda \sqrt{k}} = \frac{\sqrt{2\Omega^i_\Lambda}\mathcal{L}_\Lambda}{\mathcal{L}_{\Lambda_i}\mathcal{L}_\Lambda \sqrt{k}} = \frac{\sqrt{2\Omega^i_\Lambda}}{\sqrt{|T|}\sqrt{k}} \overset{\text{def}}{=} \frac{|E(\hat{\theta}, x^{(k)}) - F(\theta^{(k)})|}{L_k\sqrt{|T|k}} \tag{5.48b}$$

For the entropic projection update, the exact step-size (5.48a) can be computed. However, for the weighted Euclidean update, one needs a heuristic to estimate the distance between the optimal λ^* and the starting point λ^0. In this case, the current duality gap, as described in Section 5.1.3, is used. Unlike the adaptive step-size of the standard subgradient method (5.16), which depends on the whole distance, the weighted step-size (5.48b) only depends on the subset distance Ω_i of the nodes in disagreement. The duality gap should therefore be distributed evenly to all inconsistent nodes. At each iteration, if L_k denotes the number of inconsistent nodes, then the weighted adaptive step-size is naturally defined in the final expression of (5.48b).

Substituting (5.48) into (5.32) gives the explicit MD updates:

$$\rho^t = \frac{\theta_i^{t(k)}}{\sum\limits_{t\in T}\theta_i^{t(k)}}, \quad \omega^t = \text{sign}(\theta_i^{t(k)}.\bar{x}_i^{t(k)})\sqrt{2\log|T|/k}, \quad \rho^t = \frac{\rho^t\exp(\omega^t)}{\sum\limits_{t\in T}\rho^t\exp(\omega^t)}$$

$$\theta_i^{t(k+1)} = \rho^t\left(\sum_{t\in T}\theta_i^{t(k)}\right), \quad \text{for } k < k_1 \tag{5.49a}$$

$$\eta = \frac{|E(\hat{\theta}, x^{(k)}) - F(\theta^{(k)})|}{L_k\sqrt{|T|k}}$$

$$\theta_i^{t(k+1)} = \theta_i^{t(k)} + \eta\left(\bar{x}_i^{t(k)} - \frac{\sum_{t\in T}\bar{x}_i^{t(k)}}{|T|}\right), \quad \text{for } k \geq k_1 \tag{5.49b}$$

Due to the special form of the binary solutions $\bar{x}^{t(k)}$, it is straightforward to verify that the above MD updates only take place at the inconsistent nodes and labels. Furthermore, the memory requirement is no more than in the standard Euclidean updates (5.5). In the entropy sequence (5.49a), since the tree potentials θ_i^t are stored, and since each single variable $\rho_i^{t(k)}$ can be derived from the potentials, i.e. $\frac{\theta_i^{t(k)}}{\sum_{t\in T}\theta_i^{t(k)}}$, the full vector ρ need not be stored. As a result, it is only necessary to store $2 \times |T|$ additional temporary parameters ρ^t and ω^t in memory, which requires a negligible amount of additional memory. In the weighted Euclidean sequence, λ_i^t is implicitly included in the potentials update (5.49b).

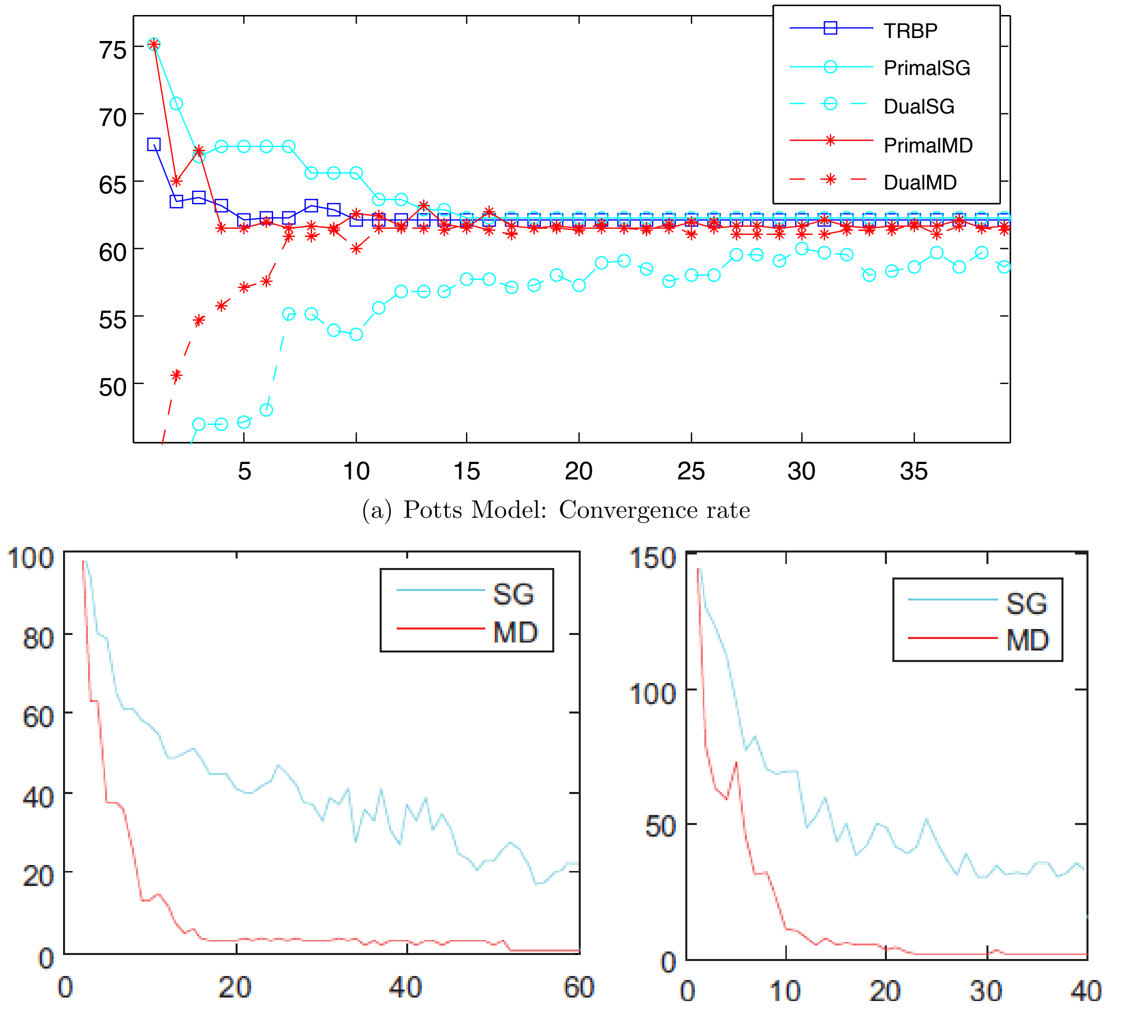
Switching criteria. An intuitive idea behind the switching criteria from entropy updates to Euclidean updates is based on the stability of the primal dual gap. When the entropy projected sequence finds a sub-optimal solution, the dual objective function will not improve further under entropy updates. Thus, the duality gap becomes stable. One can define a switching point when there is evidences of stability of the duality gap. However, subgradient type methods often show fluctuations in the objective function values, and so the primal dual gap may exhibit a corresponding zigzag behaviour. It may, therefore, take a substantial number of iterations before one can detect the stability of the duality gap. On the other hand, an important feature of the dual decomposition method for the MRF problem is that, as the method converges, the number of nodes in disagreement decreases. This observation works well in practice as no further decrease in the number of disagreement nodes after a number of iterations indicates evidence of local convergence. We define a value $\sigma < 10$ as a threshold after which to switch to Euclidean updates if there is no decrease in the number of disagreement nodes after σ iterations.

5.3 Experiments

In order to demonstrate the effectiveness of our method, experimental results with synthetic data on graph and an image segmentation problem are presented. We utilise the Undirected Graphical Models (UGM) Matlab package [93] to implement the proposed methods. In all experiments, we apply three methods: TreeReweighted Belief Propagation (TRBP), Mirror Descent (MD) with weighted distance function and the standard projected SubGradient (SG).

Algorithm 5.2: MRF Mirror Descent method $x \leftarrow \mathtt{MD}(\hat{\theta})$

```
Define a spanning tree collection $\{\theta^t\},\ t \in T$; each edge is covered exactly once;
Define the index set $\mathcal{I}$ (4.8);
Set the switching threshold $\sigma < 10$;
Set switch = false;
Set $\theta_i^{t(1)} = \hat{\theta}_i/|T|$;
for $k = 1, ..., K$ do
    for $t \in T$ do
        Compute $\bar{x}^{t(k)} \rightarrow \mathtt{BeliefPropagation}(\theta^{t(k)})$
    Find the number of disagreement nodes $L_k$;
    if $L_k \geq L_{k-1}$ then
        no-improvement = no-improvement + 1
    else
        no-improvement = 0
    if no-improvement > $\sigma$ then
        switch = true
    for $i \in \mathcal{I}$ do
        Let update = $\sum_{t \in T} \bar{x}_i^{t(k)}$;
        if $0 <$ update $< |T|$ then
            if switch = $\mathit{false}$ then
                Update the potentials $\theta^{(k+1)}$ by (5.49a)
            else
                Update the potentials $\theta^{(k+1)}$ by (5.49b)
Return $x^{(K)}$;
```



(a) Potts Model: Convergence rate

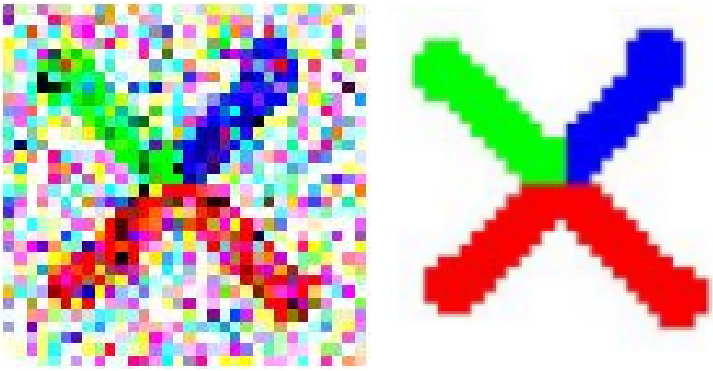
(b) Potts Model: Number of non-agreement Nodes

(c) Uniform Model: Number of non-agreement Nodes

Figure 5.1: *Synthetic data*

TRBP is a state-of-the-art method in MRF opitmisation however the global convergence are not guaranteed. Nevertheless, it is a standard practice to compare TRBP with newly developed method. In the provided UGM package, TRBP only returns the primal objective function value, while our implemented MD and SG provide both primal and dual objective function values.

Synthetic data. In the synthetic experiments, we use a grid graph of size 100 × 100 and 5 labels. For the Potts model, $\theta_{a,i}$ was drawn from $\mathcal{U}(-1,+1)$, while $\theta_{ab,ij} = \omega_{ab} * \mathbb{I}(i = j)$ and $\omega_{ab} = \mathcal{N}(0,1)$. For the uniform model, values from $\mathcal{U}(0,1)$ were assigned to the unary potentials, and values from $\mathcal{N}(0,1) * \mathcal{U}(0,1)$ for the pairwise potentials. The switching threshold was set to $\sigma = 5$. Figures 5.1(a) shows the primal-dual gap as the algorithms progress. The TRBP



(a) Corrupted image (b) Segmented image

Figure 5.2: *Image segmentation*

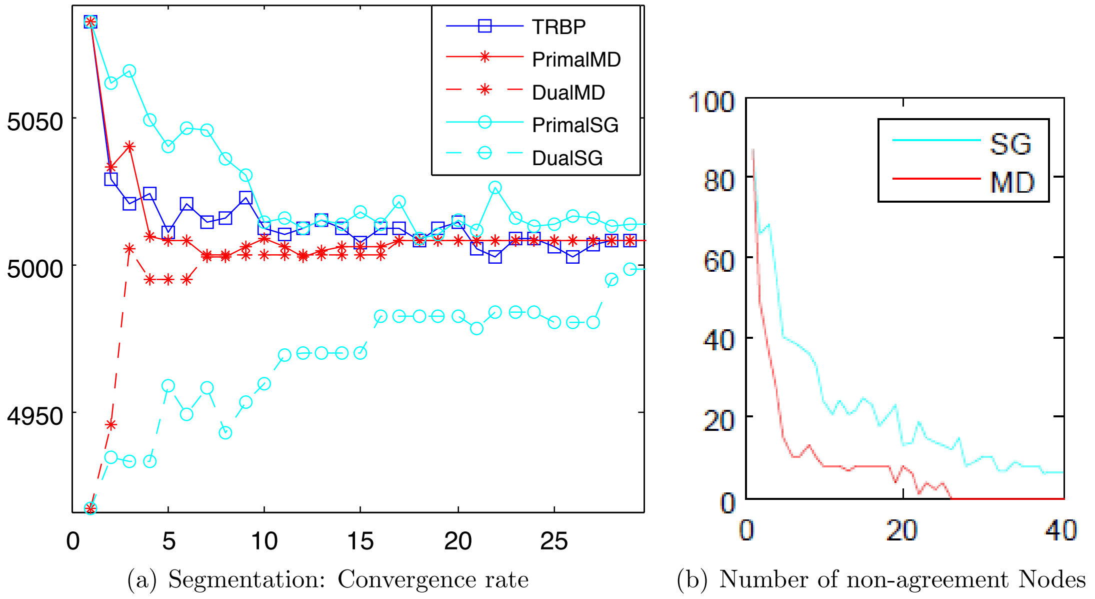
technique only computes a primal energy at each iteration and is shown for preference. The SG and MD methods compute a dual value and approximate a corresponding primal energy at each iteration. By duality theorem 4.3.1, the optimal energy is achieved when the primal-dual gap vanishes. In addition, as the primal-dual gap decreases there are less disagreement nodes and thus the number of disagreement nodes converges to zero, as shown in Figure 5.1(b). The switch to Euclidean updates occurs between iterations 20 and 25. All presented methods converge eventually, where MD outperforms SG significantly and obtains the optimal solution slightly before the TRBP method. In the Uniform model (Figure 5.1(c)), the switch to Euclidean updates does not occur as the entropy sequence is sufficient to compute the optimal labelling.

Segmentation problem. The segmentation problem aims to allocate every pixel to the best corresponding label, see Figure 5.2. There are 4 input labels: white, blue, red and green. The unary potentials are defined by the cost to assign a label $l \in L$ to a pixel $I(a)$, for example, one way of defining this cost is:

$$\theta_a(x_a = l) = -\log \rho(I(a)|a = l) \quad \forall a \in V,\ \forall l \in L$$

where $\rho(.)$ is a known probability distribution. The pairwise potentials are computed to penalise the differing label assignment of neighbouring pixels,

$$\theta_{ab}(x_a = l, x_b = k) = \exp\left(-\frac{|I(a) - I(b)|}{\sigma^2}\right) . \frac{1}{\|l - k\|} . (l \neq k) \quad \forall ab \in E,\ \forall l, k \in L$$



(a) Segmentation: Convergence rate (b) Number of non-agreement Nodes

Figure 5.3: *Image segmentation: convergence properties*

where $(l \neq k) = \{0,1\}$ and σ corresponds to the level of noise in the image. Figures 5.3(a) and 5.3(b) demonstrate the performance of the three methods. The switching step occurs between the fifteenth and twentieth iteration. Using combination of entropy and Euclidean sequence of the MD method to recover the optimal solution at around the twenty-fifth iteration.

Chapter 6

Multilevel optimisation for computer vision

In the previous chapters, we have developed specialised algorithms to solve two types of relaxations in image processing. In the first, we solve the discretised constrained image registration problem using the Sequential Quadratic Programming algorithm with a dimensional reduction technique. Adding constraints is one approach to remove undesirable solutions to ill-posed problems. The second type of relaxation is to convexify the registration problem using the Markov Random Field (MRF) model [31]. MRF is a popular model in computer vision, image and signal processing [65], machine learning and artificial intelligence. The original discrete MRF problem is NP hard and approximations are essential for computing a suboptimal solution. One popular approach is based on the linear programming (LP) relaxation (LP-MRF). However, due to the size of images, LP-MRF is intractable for standard LP solvers. This has led to the dual decomposition of LP-MRF and the development of first order methods (FOM) that exploit the structure of the dual LP-MRF. In Chapter 5, we propose a nonlinear weighted projection method based on mirror descent to accelerate the performance of standard FOMs for the dual LP-MRF. Experimental results on synthetic data and an segmentation problem show promising performance.

The above two methods take into account the image structure at a certain level of discretisation. Both the SQP algorithm (even incorporating dimensional reduction), and mirror descent (with weighted projection) cease to be of practical use when considering applications to very high dimensional problems. To overcome this difficulty, we introduce a novel approach employing different levels of discretisation for the optimisation solver. In order to reduce the computational effort needed for large problems, we propose to use low-cost steps (from a coarse discretised version of the problem). These replace some of the high cost iterations needed by the finely (or, more accurately) discretised version of the original problem, whilst ensuring global convergence. Furthermore, a coarse approximation of a finely discretised image problem could be seen as a relaxation to the problem, i.e. the coarse approximation smooths the ill-conditioned problem and reduces the non-uniqueness of solutions to the problem. As the result, using a solution of a coarse model in the fine model expects to yield an improvement step towards the true solution of the original problem.

An image problem takes infinite-dimensional images as input data. It is a common practice to discretise these images, then formulate and solve the corresponding finite-dimensional optimisation problem. The levels of discretisation lead to various finite-dimensional problems. High-(accuracy)-level discretisation provides a (approximately) true representation of the image problem but involves large data and, consequently, is computationally expensive and ill-conditioned. Low-(accuracy)-level discretisation experiences a loss in details of the images. However, it benefits from a low dimensional and smoother optimisation problem that can be relatively inexpensive to compute.

The multilevel approach is not new in image processing. Indeed, multilevel techniques for dense image registration [38], parametric registration [94] and Markov Random Fields [57] propose hierarchical discretisations for the problem where every level preserves the original structure. All these methods employ the solution of a coarse discretised problem as an initial guess for the finer optimisation problem. However, there is no established result concerning the relationship between levels of discretisation. The effect of utilising coarse models on the convergence of the overall algorithm have not been fully understood or studied.

In this chapter, we develop a general multilevel optimisation framework that employs hierarchical discretisations of an image problem. The proposed method iterates between fine and coarse levels, we establish the inter-relationship between levels of discretisations using the definition of *first order coherence*. The new algorithm is based on the proximal gradient method that can handle convex problems with simple constraints and simple nonsmooth regularisers.

6.1 Background

It is often possible to exploit the structure of large scale optimisation models to develop algorithms with lower computational complexity. A noteworthy example is the composite convex optimisation problem that consists of the minimisation of the sum of a smooth convex function and a *simple* non-smooth convex function. For a general (in contrast to *simple*) non-smooth convex function, the subgradient algorithm converges at a rate of $O(1/\sqrt{k})$, where k is the iteration number. However, if one assumes that the non-smooth component is *simple* enough such that the proximal projection step [6, 79] can be performed in closed form, then the convergence rate for function values can be improved to $O(1/k^2)$. Composite convex optimisation models arise often in a wide range of applications in computer science (e.g. machine learning), statistics (e.g the lasso problem), and engineering (e.g. signal processing), to name a few.

In addition to the composition of the objective function, many of the applications described above share another common structure. The fidelity in which the optimisation model captures the underlying application can often be controlled. Typical examples include the discretisation of partial differential equations in computer vision and optimal control [12]; the number of features in machine learning applications [105]; the number of states in a Markov Decision Processes [83]. Indeed, whenever a finite dimensional optimisation model arises from an infinite dimensional model, it is straightforward to define a hierarchy of optimisation models. In many areas it is common to take advantage of this structure by solving a low fidelity (coarse) model and using the solution as the starting point in the high fidelity (fine) model (see e.g. [38, 57] in computer vision). In this chapter, we adopt an optimisation point of view to take advantage of a hierarchy of models in a consistent manner for solving certain composite convex optimisation problems. In contrast to most multilevel methods in computer vision, we do not use the coarse model for the computation of promising starting points but rather for the computation of search directions.

The algorithm we propose is similar to the *Iterative Shrinkage Thresholding Algorithm* (ISTA) class of algorithms. There is a substantial amount of literature related to this class of algorithms and we refer the reader to [6] for a review of recent developments. The main difference between ISTA and the algorithm we propose in this chapter, is that we use both gradient information and a coarse model in order to compute a search direction. This modification of ISTA for the computation of the search direction is akin to multigrid algorithms developed recently by a number of authors. There exists a considerable number of papers exploring the idea of using multigrid methods in optimisation [12]. However the large majority of these are concerned with solving the linear system of equations to compute a search direction using linear multigrid methods (both geometric and algebraic). A different approach, and the one we adopt in this chapter is the class of multigrid algorithms proposed in [73] and further developed in [64]. The framework proposed in [73] was used for the design of a first order unconstrained line search algorithm in [112], and also related to a trust region multilevel method in [35]. The trust region approach was extended to deal with box constraints in [34]. The general constrained case was discussed in [73], but no convergence proof was given. Numerical experiments with multigrid are encouraging and a number of numerical studies have appeared so far, see e.g. [33, 74]. The algorithm we develop combines elements from ISTA and the multigrid framework developed in [73] and [112], and we call it *Multilevel Iterative Shrinkage Thresholding Algorithm* (MISTA). We prefer the name multilevel to multigrid since there is no notion of grid in our algorithm.

Past work in multilevel optimisation is largely concerned with with models where the underlying dynamics are governed by differential equations and convergence proofs exist only for the smooth case and with simple box or equality constraints. Our main contribution is the extension of the multigrid framework for convex but possibly non-smooth problems with certain types of constraints. In particular, we allow for general convex constraints, as long as the proximal projection step is computationally feasible. Apart from the work in [34] that address simple box constraints, the general case has not been addressed before. Existing approaches assume that the objective function is twice continuously differentiable, while the the proximal framework we develop in this chapter allows for a large class of non-smooth optimisation models. In addition, our convergence proof is different from the one given in [73] and [13] in that we do not assume that the algorithm used in the finest scale performs one gradient step after every coarse correction. Furthermore, our proof is based on analysing the whole sequence generated by the algorithm and does not rely on asymptotic results as in previous works [35, 112]. We show that the multilevel method using ISTA steps and coarse corrections is a *contraction* on the optimal vector with a linear convergence rate. This is the same convergence rate as ISTA. An alternative convergence analysis for ISTA is based on the reduction of function values. It has been studied in the development of FISTA [6], an accelerated version of ISTA. In term of the function value convergence, ISTA has a rate of $O(1/k)$, whilst FISTA has an improved rate of $O(1/k^2)$. The analysis of FISTA using the multilevel framework and fuction values reduction is technically more challenging currently under investigation. However, we will describe both variants of MISTA, namely MISTA-I (using ISTA steps with coarse correction) and MISTA-F (using FISTA steps with coarse correction). Despite the potentially theoretical differences between MISTA-I and FISTA, our numerical experiments show that our methods outperform both ISTA and FISTA.

6.2 Composite optimisation & quadratic approximations

In this Section we introduce our notation and the main assumptions of the proposed algorithm. We also describe the role of quadratic approximations in the design of algorithms for composite optimisation. The main difference between MISTA and existing algorithms such as ISTA and FISTA is that we do not use a quadratic approximation in all iterations. Instead we use a coarse model approximation. We describe the construction of the coarse model in Section 6.3.2. In Section 6.3.4 we provide a motivating example to explain why a quadratic approximation may be inferior to a coarse approximation for certain classes of problems.

6.2.1 Notation and problem description

We will assume that the optimisation model can be formulated using only two levels of fidelity, a fine model and a coarse model. We use h and H to indicate whether a particular quantity/property is related to the fine and coarse model respectively. It is easy to generalize the algorithm to more levels but with only two levels the notation is simpler. The fine model is the convex composite optimisation model,

$$\min_{x_h \in \Omega_h} \left\{ F_h(x) \triangleq f_h(x_h) + \lambda g_h(x_h) \right\}, \tag{6.1}$$

where $\Omega_h \subset \mathbb{R}^h$ is a closed convex set, f_h is a smooth function with a Lipschitz continuous gradient, and $g_h : \mathbb{R}^h \to \mathbb{R}$ is an extended value convex function that is possibly non-smooth. When g_h is a norm then the scalar $\lambda \geq 0$ is a regularisation parameter, and so the non-smooth term encourages solutions that are sparse. Sparsity is a desirable property in many applications. The algorithm we propose does not only apply when g_h is a norm. But if it is a norm, then some variants of our algorithm make use of the dual norm associated with g_h and so without loss of generality we assume that λ is given (in general it can be taken to be 1). We use L_h to denote the Lipschitz constant of the gradient of f_h. The incumbent solution at iteration k in resolution h is denoted by $x_{h,k}$. We use $f_{h,k}$ and $\nabla f_{h,k}$ to denote $f_h(x_{h,k})$ and $\nabla f_h(x_{h,k})$ respectively.

6.2.2 Quadratic approximation and ISTA

The prevailing way to update $x_{h,k}$ is to perform a quadratic approximation of the smooth component of the objective function, and then solve the following *proximal subproblem*,

$$x_{h,k+1} = \arg\min_{y\in\Omega_h} f_{h,k} + \langle \nabla f_{h,k}, y - x_{h,k}\rangle + \frac{L_h}{2}\left\| x_{h,k} - y\right\|^2 + g(y).$$

Note that the above can be rewritten as follows,

$$x_{h,k+1} = \arg\min_{y\in\Omega_h} \frac{L_h}{2}\left\| y - \left(x_{h,k} - \frac{1}{L_h}\nabla f_{h,k}\right)\right\|^2 + g(y).$$

When the Lipschitz constant is known, ISTA keeps updating the solution vector by solving the optimisation problem above [6]. Another example is the classical gradient projection algorithm[49], in this case the proximal projection step is given by,

$$\min_{y\in\mathbb{R}^h} \frac{L_h}{2}\left\| y - \left(x_{h,k} - \frac{1}{L_h}\nabla f_{h,k}\right)\right\|^2 + I_{\Omega_h}(y),$$

where I_{Ω_h} is the indicator function on Ω_h. For later use we define the generalized *proximal operator* as follows,

$$\text{prox}_h(x) = \arg\min_{y\in\Omega_h} \frac{1}{2}\left\| y - x\right\|_2^2 + g(y). \tag{6.2}$$

Our algorithm uses the step-size differently than ISTA/FISTA and so in proximal steps the step-size does not appear explicitly in the definition of the proximal projection problem. Our proximal update step is given by,

$$x_{h,k+1} = x_{h,k} - s_{h,k}D_{h,k} \tag{6.3}$$

where the *gradient mapping* $D_{h,k}$ is defined as follows,

$$D_{h,k} \triangleq \left[x_{h,k} - \text{prox}_h(x_{h,k} - \frac{1}{L_h}\nabla f_{h,k})\right] \tag{6.4}$$

Updating the incumbent solution in this manner is reminiscent of classical gradient projection algorithms [91].

When g_h is a norm we will also make use of the properties of the *dual norm proximal operator* defined as follows,

$$\begin{aligned} \text{proj}_h^*(x) = \arg\max_y \; & -\frac{1}{2}\left\|y - x\right\|_2^2 - \left\|x\right\|^2 \\ \text{s.t. } & g^*(y) \leq \lambda, \end{aligned}$$

where g^* is the dual norm of g. Using Fenchel duality (see Lemma 2.3 in [98]) it can be shown that,

$$\text{prox}_h(x) = x - \text{proj}_h^*(x). \tag{6.5}$$

The relationship above is used to compute the proximal projection step efficiently.

6.3 Multilevel ISTA (MISTA)

Rather than always construct a quadratic approximation, we propose to construct an approximation with favorable computational characteristics for at least some iterations. Favorable computational characteristics in the context of optimisation algorithms may mean reducing the dimensions of the problem and possibly increasing the smoothness of the model. This approach facilitates the use of non-linear (but still convex) approximations around the current point. The motivation behind this class of approximations is that the global nature of the approximation would reflect global properties of the model that would yield better search directions. A motivating example that makes this idea concrete is given in Section 6.3.4.

There are three components to the construction of the proposed algorithm: (a) specification of the restriction/prolongation operators that transfer information between different levels; (b) construction of an appropriate hierarchy of models; (c) specification of the algorithm (smoother) to be used in the coarse model. Below we address these three components in turn.

6.3.1 Information transfer between levels

Multilevel algorithms require information to be transferred between levels. In the proposed algorithm we need to transfer information about the incumbent solution, proximal projection and gradient around the current point. At the fine level the design vector x_h is a vector in $\mathbb{R}^h$. At the coarse level the design vector is a vector in $\mathbb{R}^H$ and $H < h$. At iteration k, the proposed algorithm projects the current solution $x_{h,k}$ from the fine level to coarse level to obtain an initial point for the coarse model denoted by $x_{H,0}$. This is achieved using a suitably designed matrix (I_h^H) as follows,

$$x_{H,0} = I_h^H x_{h,k}.$$

The matrix $I_h^H \in \mathbb{R}^{H \times h}$, is called a *restriction operator* and its purpose is to transfer information from the fine to the coarse model. There are many ways to define this operator and we will discuss some possibilities for machine learning problems in Section 6.5. This is a standard technique in multigrid methods both for solutions of linear and nonlinear equations and for optimisation algorithms [20, 73]. In addition to the restriction operator we also need to transfer information from the coarse model to the fine model. This is done using the *prolongation operator* $I_H^h \in \mathbb{R}^{h \times H}$. The standard assumption in multigrid literature [20] is to assume that $I_h^H = c(I_H^h)^\top$, where c is some positive scalar. With out loss of generality we will assume that $c = 1$.

6.3.2 Coarse model construction

The construction of the coarse models in multilevel algorithms is a subtle process. It is this process that sets apart rigorous multilevel algorithms with performance guarantees from heuristic approaches (e.g. kriging methods) used in the engineering literature. A key property of the coarse model is that locally (i.e. at the initial point of the coarse model, $x_{H,0}$) the optimality conditions of the two models match. In the unconstrained case this is achieved by adding a linear term in the objective function of the coarse model [35, 73, 112]. In the constrained case the linear term is used to match the gradient of the Lagrangian [73]. However, the theory for the constrained case of multilevel algorithms is less developed. Here we propose an approach that contains the unconstrained approach in [73] and the box-constrained case [34] as special cases. In addition we are able to deal with the non–smooth case and through the proximal step we address the constrained case.

In the case where the optimisation model is non–smooth there many ways to construct a coarse model. We propose three ways to address the non–smooth part of the problem. All three approaches enjoy the same convergence properties, but depending on the application some coarse models may be more appropriate since they make different assumptions regarding the non–smooth function and the prolongation/restriction operators. The three approaches are: (a) smoothing the non–smooth term, (b) a reformulation using dual norm projection, (c) non–smooth model with a projection using the indicator function. The coarse model in all three approaches has the following form,

$$F_H(x_H) \triangleq f_H(x_H) + g_H(x_H) + \langle v_H, x_H \rangle. \tag{6.6}$$

We assume that given the function f_h, the construction of f_H is easy (e.g. varying a descretisation parameter or the resolution of an image etc.) and has Lipschitz continuous gradients. The second term in (6.6) represents information regarding the non–smooth part of the original objective function, and the third term ensures the fine and coarse model are coherent (in the sense of Lemmas 6.3.1-6.3.3). We will denote the smooth part of the objective function with,

$$\phi_H(x_H) \triangleq f_H(x_H) + \langle v_H, x_H \rangle.$$

Clearly, the differentiable part of the objective has the same Lipschitz continuous gradient as f_H,

$$\|\nabla \phi_H(x_H) - \nabla \phi_H(y_H)\| \leq L_H \|x_H - y_H\|.$$

Apart from f_H, the other two terms in (6.6) vary depending on which of the three approaches is adopted. We discuss the three options in decreasing order of generality below.

The smooth coarse model.

The approach that requires the least assumptions about the model is to construct a coarse model by smoothing the non–smooth part of the objective function. In other words, the second term in (6.6) is again a reduced order version of g_h but is also smooth. In the application we consider the non-smooth term is usually a norm or an indicator function. It is therefore easy to construct a reduced order version of g_h, and there exists many methods to smooth a non–smooth function [7]. Our theoretical results do not depend on the choice of the smoothing method. We construct the last term in (6.6) with,

$$v_H = L_H I_h^H D_{h,k} - (\nabla f_{H,0} + \nabla g_{H,0}). \tag{6.7}$$

When the coarse model is smooth, then L_H corresponds to the Lipschitz constant of (6.6).

Lemma 6.3.1 *Suppose that f_H and g_H have Lipschitz continuous gradients, and that the coarse model associated with* (6.1) *is given by,*

$$\min_{x_H} \; f_H(x_H) + g_H(x_H) + \langle v_H, x_H \rangle, \tag{6.8}$$

where v_H is given by (6.7)*, then,*

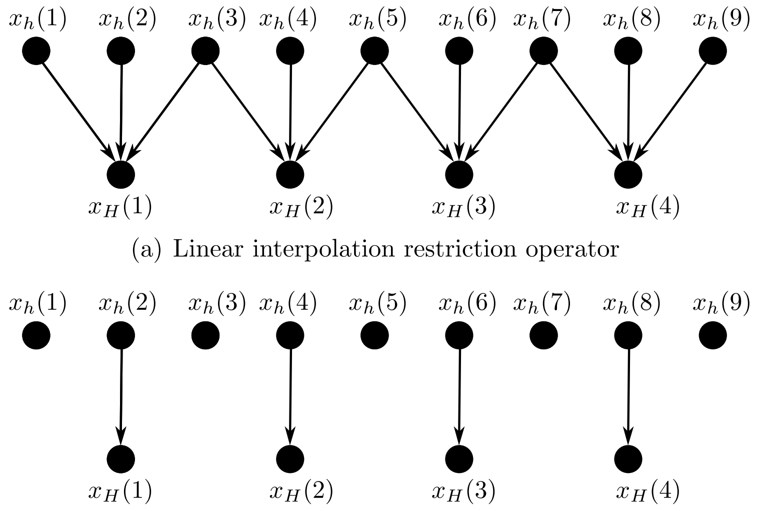
$$D_{H,0} = I_h^H D_{h,k}. \tag{6.9}$$

Proof Using the definitions of the gradient mapping in (6.4) and the projection operator (instead of the prox operator) for the smooth objective function of the coarse level, we obtain:

$$\begin{aligned}
D_{H,0} &= x_{H,0} - \mathrm{prox}_H(x_{H,0} - \frac{1}{L_H}\nabla F_{H,0}) \\
&= x_{H,0} - \arg\min_{z \in \mathbb{R}^H} \frac{1}{2}\|z - \left(x_{H,0} - \frac{1}{L_H}\nabla F_{H,0}\right))\|^2 \\
&= \frac{1}{L_H}\nabla F_{H,0} \\
&= \frac{1}{L_H}(\nabla f_{H,0} + \nabla g_{H,0} + v_H) \\
&= I_h^H D_{h,k},
\end{aligned}$$

where in the second equality we used the fact that the objective function in (6.8) is smooth and so any constraints in the form of $x_H \in \Omega_H$ can be incorporated in g_H. ∎

The condition in (6.9) is referred to as the *first order coherent condition.* It ensures that at if $x_{h,k}$ is optimal in the fine level, then $x_{H,0} = I_h^H x_{h,k}$ is optimal in the coarse model. This property is crucial in establishing convergence of multilevel algorithms. The smooth case was discussed in [35, 73, 112], and the Lemma above extends the condition to the non-smooth case.



(a) Linear interpolation restriction operator

(b) Coordinate wise restriction operator

Figure 6.1: *(a) The linear interpolation operator widely used in the multigrid literature. (b) The coordinate wise restriction operator is reminiscent of the techniques used in coordinate descent algorithms.*

Next we discuss a different way to construct the coarse model (and hence a different v_H term) that makes a particular assumption about the restriction and interpolation operators.

A non-smooth coarse model with dual norm projection.

In the coarse construction method described above we imposed a restriction on the coarse model but allowed arbitrary restriction/prolongation operators. In our second method for constructing coarse models we allow for arbitrary coarse models (they can be non-smooth) but make a specific assumption regarding the information transfer operators. In particular we assume that,

$$x_H(i) = (I_h^H x_h)_i = x_h(2i), \quad i = 1, \ldots, H.$$

We refer to this operator as a *coordinate wise restriction operator*. The reason we discuss this class of restriction operators is that in the applications we consider the non-smooth term is usually a norm that satisfies the following,

$$\text{proj}_H^*(I_h^H x_h) = I_h^H \text{proj}_h^*(x_h), \tag{6.10}$$

where proj_h^* and proj_H^* denote projection with respect to the dual norm associated with g_h and g_H respectively. When the restriction operator is done coordinate wise then the preceding equation is satisfied for many frequently encountered norms including the l_1, l_2 and l_∞ norms used as regularisers. In our second coarse construction method the last term in (6.6) is constructed with,

$$v_H = \frac{L_H}{L_h} I_h^H \nabla f_{h,k} - \nabla f_{H,0}. \tag{6.11}$$

Lemma 6.3.2 *Suppose that f_H has a Lipschitz continuous gradient, condition* (6.10) *is satified, and that both g_h and g_H are norms. For the coarse model associated with* (6.1) *given by,*

$$\min_{x_H} \; f_H(x_H) + g_H(x_H) + \langle v_H, x_H \rangle, \tag{6.12}$$

where v_H is given by (6.11), *then,*

$$D_{H,0} = I_h^H D_{h,k}. \tag{6.13}$$

Proof Since g_h is a norm, we can compute the proximal term by (6.5) to obtain,

$$\begin{aligned} D_{h,k} &= \left[x_{h,k} - \mathrm{prox}_h \left(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k} \right) \right] \\ &= \left[x_{h,k} - \left(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k} - \mathrm{proj}_h^* \left(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k} \right) \right) \right] \\ &= \frac{1}{L_h} \nabla f_{h,k} + \mathrm{proj}_h^* \left(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k} \right). \end{aligned}$$

Using the same argument for the coarse model and the definition in (6.11),

$$\begin{aligned} D_{H,0} &= \frac{1}{L_H} \left(\nabla f_{H,0} + v_H \right) + \mathrm{proj}_H^* \left(x_{H,0} - \frac{1}{L_H} \left(\nabla f_{H,0} + v_H \right) \right) \\ &= I_h^H \left(\frac{1}{L_h} \nabla f_{h,k} \right) + \mathrm{proj}_H^* \left(I_h^H \left(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k} \right) \right) \\ &= I_h^H \left(\frac{1}{L_h} \nabla f_{h,k} + \mathrm{proj}_h^* (x_{h,k} - \frac{1}{L_h} \nabla f_{h,k}) \right) \\ &= I_h^H D_{h,k}. \end{aligned}$$

Where in the third equality we used (6.10). ∎

Next we discuss a different way to construct the coarse model (and hence a different v_H term) that makes a particular assumption on the non-smooth component of the fine model.

A non-smooth coarse model with constraint projection.

When the non-smooth term is a regularization term, the proximal term is computationally tractable. In this case, the problem can equivalently be formulated using a constraint as opposed to a penalty term. In this third method for constructing coarse models we assume that the coarse non-smooth term is given by,

$$g_H(x_H) = \begin{cases} x_H \text{ if } x_H \in \Omega_H, \\ \infty \text{ otherwise.} \end{cases}$$

With this definition, the coarse model has the same form as in (6.6) where g_H is an indicator function on Ω_H, and the final term is constructed using the following definition for v_H,

$$v_H = L_H x_{H,0} - \nabla f_{H,0} - L_H I_h^H \text{prox}_h(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k}). \tag{6.14}$$

We also make the following assumption regarding the relationship between coarse and fine feasible sets,

$$\text{proj}_H(I_h^H x_h) = I_h^H x_h, \ \forall x_h \in \Omega_h. \tag{6.15}$$

The condition above is satisfied for many situations of interest, for example when $\Omega_h = \mathbb{R}_+^h$ and $\Omega_H = \mathbb{R}_+^H$. It also holds for box constraints and simple linear or convex quadratic constraints. If the condition above is not possible to verify then the other two methods described in this section can still be used. Note that we only make this assumption regarding the coarse model, i.e. we do not require such a condition to hold when we prolong feasible coarse models to the fine model.

Lemma 6.3.3 *Suppose that that condition* (6.15) *is satisfied, f_H has a Lipschitz continuous gradient and that g_H is an indicator function on $\Omega_H \subset \mathbb{R}^H$. Assume that the coarse model associated with* (6.1) *is given by,*

$$\min_{x_H} \ f_H(x_H) + g_H(x_H) + \langle v_H, x_H \rangle, \tag{6.16}$$

where v_H is given by (6.14)*, then*

$$D_{H,0} = I_h^H D_{h,k}. \tag{6.17}$$

Proof Using the fact that the proximal step in the coarse model reduces to an orthogonal projection on Ω_H we obtain,

$$\begin{aligned} D_{H,0} &= x_{H,0} - \mathrm{proj}_H(x_{H,0} - \frac{1}{L_H}(\nabla f_{H,0} + v_H)) \\ &= x_{H,0} - \mathrm{proj}_H(I_h^H \mathrm{prox}_h(x_{h,k} - \frac{1}{L_h}\nabla f_{h,k})) \\ &= I_h^H \left[x_{x,k} - \mathrm{prox}_h(x_{h,k} - \frac{1}{L_h}\nabla f_{h,k}) \right] \\ &= I_h^H D_{h,k}, \end{aligned}$$

where in the third equality we used assumption (6.15). ∎

6.3.3 Algorithm description

In the previous section we described ways to construct a coarse model, and specified the information transfer operators. Given these two components we are now in a position to describe the algorithm in full. It does not matter how the coarse model or the information transfer operators were constructed. The only requirement is that the first order coherence condition is satisfied (Lemmas 6.3.1, 6.3.2, 6.3.3). It is important to impose this condition in order to be able to prove that the algorithm converges. However, it does not matter how this condition is imposed in the coarse mode. The prolongation/restriction operators are also satisfy assumed to $I_h^H = c(I_H^h)^\top$ for some constant $c > 0$. The latter assumption is standard in the literature of multigrid methods.

Given an initial point $x_{H,0}$, the coarse model is solved in order to obtain a so called *error correction term.* The error correction term is the vector that needs to be added to the initial point of the coarse model in order to obtain an optimal solution $x_{H,\star}$ in (6.6),

$$e_{H,\star} = x_{H,0} - x_{H,\star}.$$

In practice the error correction term is only approximately computed, and instead of $e_{H,\star}$ we will use $e_{H,m}$, i.e. the error correction term after m iterations. After the coarse error correction

Algorithm 6.1: Multilevel Iterative Shrinkage Thresholding Algorithm

```
if Condition to restrict current iterate $x_{h,k}$ to coarse model is satisfied then
    Set $x_{H,0} = I_h^H x_{h,k}$;
    Compute $m$ iterations of the coarse level
        $$x_{H,m} = x_{H,0} + \sum_{i=0}^{m} s_{H,i} D_{H,i}$$
    Set $d_{h,k} = I_H^h (x_{H,0} - x_{H,m})$;
    Find a suitable $\tau$ that satisfies (6.44), compute:
        $$x^+ = \mathrm{prox}_h(x_{h,k} - \tau d_{h,k})$$
    Choose a step-size $s \in (0,1]$ that satisfies (6.41) to update:
        $$x_{h,k+1} = x_{h,k} - s(x_{h,k} - x_h^+) \qquad (6.18)$$
else
    Compute gradient mapping:
        $$D_{h,k} = x_{h,k} - \mathrm{prox}_h(x_{h,k} - \frac{1}{L_h} \nabla f_{h,k})$$
    Choose a step-size $s \in (0,1]$ to update:
        $$x_{h,k+1} = x_{h,k} - s D_{h,k} \qquad (6.19)$$
```

term is computed, it is projected to the fine level using the prolongation operator:

$$d_{h,k} = I_H^h e_{H,m} \triangleq I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i}$$

and used to update the current solution,

$$x_{h,k+1} = x_{h,k} - s_{h,k}(x_{h,k} - x_h^+),$$

where $s_{H,i}, s_{h,k}$ are appropriate stepsizes and,

$$x_h^+ = \text{prox}_h(x_{h,k} - \tau d_{h,k})$$

Clearly, if $d_{h,k} = \nabla f_{h,k}$, $\tau = 1/L_H$, then the algorithm performs exactly the same step as ISTA with the proximal update step given in (6.3). Below we specify a conceptual version of the algorithm. Anticipating the generalization of MISTA to multiple levels we introduce the notation,

$$\phi_l(x_l) = f_l(x_l) + \langle v_l, x_l \rangle, \quad l = h, H.$$

to denote the smooth part of the objective function at level l. If $l = h$ then $v_l = 0$.

There are many choices we need to make before we can obtain a fully implementable algorithm, including a parameter to choose when to perform a search in the coarse level, the stepsizes to be used in the different levels, the number of iterations in the coarse level and so on. Most of these parameters will be defined by the convergence analysis in Section 6.4. However, before we discuss the theoretical properties of the algorithm we present a motivating example to illustrate why we expect this algorithm to perform well.

6.3.4 Motivating example

There are two reasons why the algorithm described above could yield good results. Firstly, the coarse model is a lower dimensional model which means that one could use an algorithm that has superior convergence properties, e.g. one could use a first order algorithm in the fine level and a second order algorithm in the coarse level. In addition coarsening in many applications has the effect of reducing the Lipschitz constant and therefore less iterations are required to obtain a solution. Thus for both theoretical and practical reasons one would expect to solve the coarse model in fewer iterations than the coarse model. Secondly, the coarse model is in some sense a global approximation of the original model one would expect the improvement due to the correction term to have significant impact to the solution of the fine model. However this latter benefit depends on how much information is lost due to coarsening. We illustrate these two points using two simple examples.

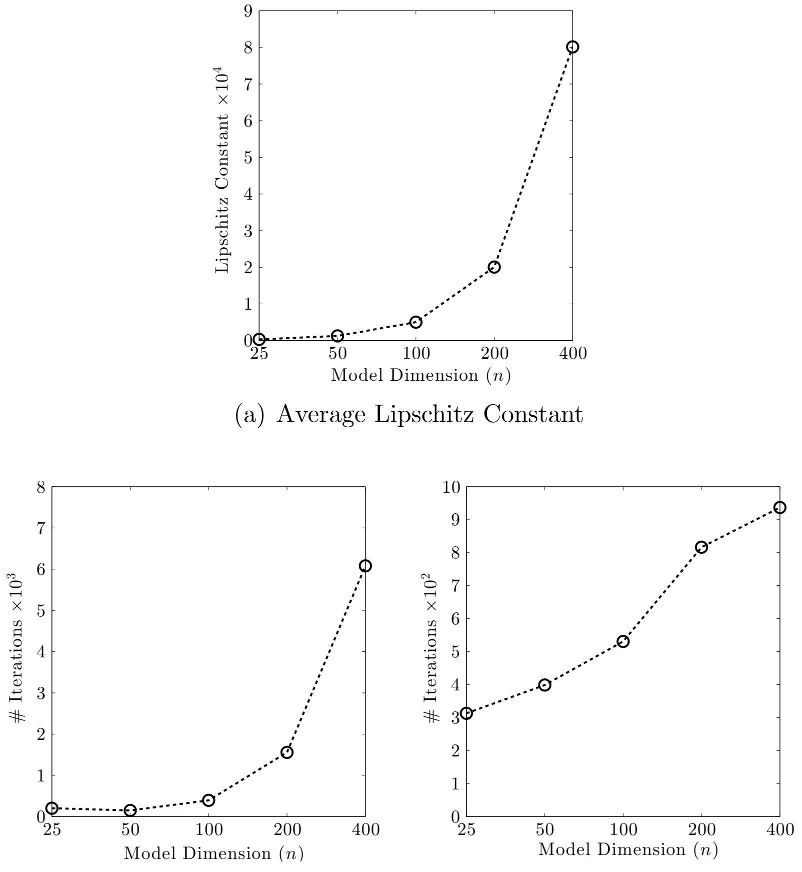
Consider the following problem arising in linear inverse models,

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2^2 + \mu\|x\|_p,$$

where $b \in \mathbb{R}^m$, $A \in \mathbb{R}^m \times n$, $p \in \{1, 2, \infty\}$ and $\mu > 0$ is the regularisation constant. This class of models arise in computer vision, machine learning, and in numerous applications in statistics. To illustrate the effect of the number of dimensions to the Lipschitz constant and the convergence rate of first order algorithms we randomly created 10^4 models for $n = 25,\ 50,\ 100,\ 200,$ and 400. The matrix A and b were created randomly, and we took $m = n$ in our experiments. We solved all models (using $p = 1$)within 1% of the optimal solution. We report average results in Figure 6.2. The standard error associated with the average is very small for all cases (less than 0.03% of the mean). In Figure 6.2(a) we plot the Lipschitz constant associated with the model (for this application the Lipschitz constant is given by $2\lambda_{\max}(A^\top A)$). As is expected the Lipschitz constant grows with the number of dimensions. The effect of the Lipschitz constant is to increase the number of iterations for both ISTA (Figure 6.2(b)) and FISTA (Figure 6.2(c)). Secondly, a simple image restoration demonstrates the expected benefit due to the global nature of the coarse approximation. The aim of the application is to recover an original image from a corrupted blurry image. The application is not only used for images but also applicable for video or audio signals, and is formulated as the following optimisation model,

$$\min_{x\in\Omega} \|Ax-b\|_2^2 + \lambda\|W(x)\|_1, \tag{6.20}$$

where b denotes the original image, and $W(x)$ is wavelet transformation of the image. The first term aims to find an image that is as close to the original image as possible, and the second term enforces a relationship between the pixels and ensures the recovered image is smooth. Note that the first term is convex and differentiable, the second term is also convex but non-smooth. This problem fits exactly the framework of convex composite optimisation. In addition it is



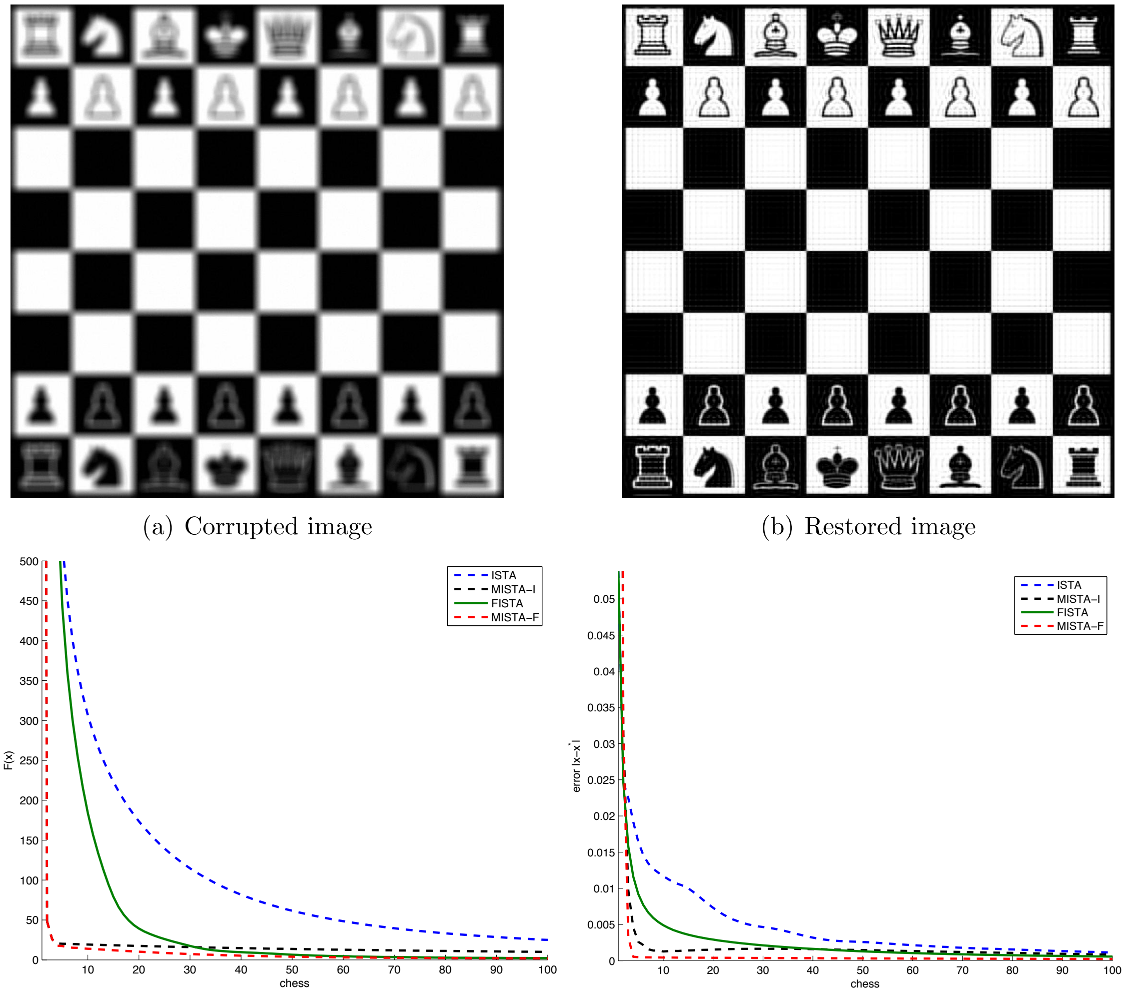
(a) Average Lipschitz Constant

(b) ISTA Performance

(c) FISTA Performance

Figure 6.2: *(a) Average Lipschitz constant for randomly generated linear inverse models grows with the number of dimensions (b) Number of iterations for the ISTA. (c) Number of iterations for the FISTA. When comparing the performance of ISTA and FISTA note the change of scale in the y-axis. In both cases the number of iterations required to reach within* 1% *of the optimal solution grows with the number of dimensions.*

easy to define a hierarchy of models by varying the resolution of the image. The restriction operator to generate coarse models for the image restoration problem is discussed in Section 6.5. Consider the structured image in Figure 6.3(a), and note that not much information is lost if its resolution is reduced. Indeed for this image, our algorithm clearly outperforms both ISTA and the theoretically superior FISTA. In Figure 6.3(c) we plot the function value obtained from the three algorithms and note that the MISTA algorithm obtains the optimal solution from nearly the first iteration. It is also worth noting that all three algorithms were initialized using the input corrupted image. At the first few iterations of ISTA/FISTA, proximal gradient updates are used. These updates do not provide significant improvement compared to the first coarse correction that is used by MISTA-I/F. From (6.3(d)) we see that in term of the optimal solution vector, MISTA-I/F are clearly superior to the other two. The reason why the algorithm works so well in this case is obvious: coarsening this simple image does not change the image in a substantial manner. Of course not all images are as simple as the one in this example but still all images have some sort of structure that can be exploited. We will consider much more complicated images and different computer vision applications in Section 6.5. While it is unrealistic to expect the algorithm to be so much better in more complicated models, we still show that there are clear advantages of MISTA compared to the state of the art.



(a) Corrupted image

(b) Restored image

(c) Function value

(d) Distance from optimum vector $x^\star$

Figure 6.3: *(a) A simple image that does not loose any information by reducing its resolution, (b) Same as the image in (a) but with noise, (c) Comparison of the three algorithms in terms of function value, (d) Distance from optimum vector $x^\star$.*

6.4 Global convergence rate analysis

The convergence of MISTA is studied in this section. We show that MISTA converges linearly to the optimal solution. Let us denote by $x_{h,*}$ the optimal solution of the composite function (6.1). Since the objective function is strictly convex, $x_{h,*}$ is unique. The convergence rate of MISTA is derived by showing that it is a contraction algorithm on the optimal solution vector $x_{h,*}$. There are two issues to address, the contraction of gradient-mapping steps (6.19), and the contraction of coarse correction steps (6.18). To this end, we provide a proof based on the contraction principle:

$$\|x_{h,k+1} - x_{h,*}\|^2 \leq \sigma \|x_{h,k} - x_{h,*}\|^2 \tag{6.21}$$

where $\sigma < 1$ is a contraction modulus, $x_{h,k+1}$ is a result of an update from either the gradient mapping (6.19), or the coarse correction (6.18). The contraction property of the gradient mapping is similar to the Fejer-monotonicity of projection method, when setting $s = 1$. In this section, we do not analyse the contraction property of proximal gradient update (6.19), as that has been considered in [100, 19] and references therein. The following lemma follows from Theorem 3.4 and section 3.3 in [100]; or Proposition 2 and Remark 7 in [19]. The lemma establishes the contraction property when $x_{h,k+1}$ is generated by the gradient mapping,

Lemma 6.4.1 *Suppose $s \in (0, 1]$, then the gradient mapping iteration (6.19) is a contraction:*

$$\|x_{h,k+1} - x_{h,*}\|^2 \leq \sigma_{ISTA} \|x_{h,k} - x_{h,*}\|^2 \tag{6.22}$$

where $\sigma_{ISTA} \in (0, 1)$ is a contraction modulus.

We derive below the conditions ensuring the convergence of MISTA. The key considerations are the contraction property of the gradient proximal step (Lemma 6.4.1) and the coarse correction step.

For ease of presentation, we use $\|.\|$ to denote $\|.\|_2$ in the rest of the chapter. The expansion of the left side of inequality (6.21) leads to the following two terms:

$$\langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*} \rangle \tag{6.23}$$

$$\|d_{h,k} - d_{h,*}\|^2 \tag{6.24}$$

These two terms need to be related to the right side of (6.21). A fundamental property of the coarse correction is based on the local equivalence between the coarse and fine levels of the gradient mapping. This local equivalence is ensured by the first order coherence property (Lemmas 6.3.1, 6.3.2, 6.3.3),

$$D_{H,0} = I_h^H D_{h,k} \tag{6.25}$$

In addition to the convexity of the composite function, we invoke the following weak assumptions.

Assumption 6.4.2 *For a given pair of restriction/prolongation operators, there exist two constants $\omega_1 \geq 1$ and $\omega_2 \leq 1$, such that:*

$$\|I_h^H y_h\| \leq \omega_1 \|y_h\| \tag{6.26a}$$

$$\|I_H^h y_H\| \leq \omega_2 \|y_H\| \tag{6.26b}$$

for any vectors y_h in the fine level, and y_H in the coarse level.

The above assumptions are indeed satisfied by most common restriction/prolongation operators. For example, let us consider two common restriction operators as illustrated graphically in Figure 6.1. For the linear interpolation operator I_h^H, assume that we have a fine vector $y_h \in \mathbb{R}^6$ and a coarse vector $y_H \in \mathbb{R}^2$. The operator I_h^H groups c fine nodes for every corresponding coarse node (in Fig 6.1(a), $c = 3$) as defined (in this case) by the matrix:

$$I_h^H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

then the prolongation operator is given by $I_H^h = \frac{1}{c}(I_h^H)^\top$. Clearly, we can set:

$$\omega_1 = \|I_h^H\| = \max_{y_h \neq 0} \frac{\|I_h^H y_h\|}{\|y_h\|} = \sqrt{c} \geq 1, \forall y_h \neq 0$$

as, always, $c \geq 1$. On the other hand,

$$\omega_2 = \|I_H^h\| = \frac{1}{\sqrt{c}} \max_{y_H \neq 0} \frac{\|(I_h^H)^\top y_H\|}{\|y_H\|} = 1, \forall y_H \neq 0$$

For the coordinate wise operator, assume that we work with $y_h \in \mathbb{R}^4$ and $y_H \in \mathbb{R}^2$, and the nodes with odd indices are omitted in the coarse vector. So, the restriction operator is defined as,

$$I_h^H = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the prolongation is simply given by $I_H^h = (I_h^H)^\top$. Then the upper bounds of ω_1, ω_2 are:

$$\begin{aligned} \omega_1 &= \|I_h^H\| = \max_{y_h \neq 0} \frac{\|I_h^H y_h\|}{\|y_h\|} = 1 \quad \text{when } y_h(2i+1) = 0,\ i = 0, ..., h \\ \omega_2 &= \|I_H^h\| = \max_{y_H \neq 0} \frac{\|(I_h^H)^\top y_H\|}{\|y_H\|} = 1, \forall y_H \neq 0 \end{aligned}$$

Additionally, the optimality condition for the proximal type algorithm can be found in [9], and given by the following lemma.

Lemma 6.4.3 *At the optimal solution $x_{h,*}$, we have:*

$$x_{h,*} = x_{h,*} - sD_{h,*}$$

for any $s > 0$. Indeed, the gradient mapping at the optimal solution satifies

$$D_{h,*} = 0. \tag{6.27}$$

The restriction of the optimal vector at a fine level leads to a stationary point at a coarse level. This is straightforward to derive in the following corollary,

Corollary 6.4.4 *Let $x^*_{H,0}$ denote a coarse model of the optimal solution vector $x_{h,*}$ at the fine level (i.e $x^*_{H,0} = Rx_{h,*}$). Then the coarse model satisfies*

$$D^*_{H,i} = 0 \ , \ \forall i$$

.

Based on the these stationary properties at the fine and the coarse levels, we can estabish the conditions for efficient coarse corrections,

$$
\begin{aligned}
\|I_h^H D_{h,k}\| &> \kappa \|D_{h,k}\| && (6.28)\\
\|x_h^k - \tilde{x}_h\| &> \eta \|\tilde{x}_h\|. && (6.29)
\end{aligned}
$$

As we see from the first order coherence property (6.25), $I_h^H D_{h,k}$ equals to the gradient-mapping of the coarse level. Therefore, the condition (6.28) prevents the method from solving the coarse level when its first order optimality is almost achieved. The current iterate appears to be a stationary point for the coarse model and it will not improve in the coarse subspace. Typically, κ is the tolerance on the norm of the first-order optimality condition of (the fine) level h or alternatively $\kappa \in (0, \min(1, \min \|I_h^H\|))$. In condition (6.29), $\tilde{x}_h$ is recorded as the latest point generated by the corresponding coarse correction. A new coarse correction should not be used when the current point is very close to $\tilde{x}_h$. The motivation is that performing a coarse correction at a point x_h^k that satisfies both the above conditions will yield a new point close to the current x_h^k.

In order to establish the relationship between the two terms (6.23) and (6.24) with the right side of inequality (6.21), we need the following key property of the gradient mapping [3, Lemma 2.3].

Lemma 6.4.5 *Consider a discretised (coarse or fine) level. For two arbitrary vectors $x, y \in \Omega$, let the smooth function ϕ have L-lipschitz continuous gradients, and D_x, D_y gradient mappings, then:*

$$\langle D_x - D_y, x - y\rangle \geq \frac{3}{4}\|D_x - D_y\|^2. \tag{6.30}$$

Proof The proximal operator is known to be firmly nonexpansive, i.e.

$$
\begin{aligned}
&\left\langle \text{prox}(x - \frac{1}{L}\nabla\phi_x) - \text{prox}(y - \frac{1}{L}\nabla\phi_y), (x - \frac{1}{L}\nabla\phi_x) - (y - \frac{1}{L}\nabla\phi_y)\right\rangle \\
\geq &\left\|\text{prox}(x - \frac{1}{L}\nabla\phi_x) - \text{prox}(y - \frac{1}{L}\nabla\phi_y)\right\|^2.
\end{aligned}
$$

Using the definition $D_x = [x - \text{prox}(x - \frac{1}{L}\nabla\phi_x)]$ to get:

$$\left\langle (x - D_x) - (y - D_y), (x - \frac{1}{L}\nabla\phi_x) - (y - \frac{1}{L}\nabla\phi_y) \right\rangle \geq \|(x - D_x) - (y - D_y)\|^2$$

which is equivalent to:

$$\left\langle (x - D_x) - (y - D_y), (D_x - \frac{1}{L}\nabla\phi_x) - (D_y - \frac{1}{L}\nabla\phi_y) \right\rangle \geq 0$$

Therefore:

$$\begin{aligned} \langle D_x - D_y, x - y\rangle \geq \; & \|D_x - D_y\|^2 + \tfrac{1}{L}\langle \nabla\phi_x - \nabla\phi_y, x - y\rangle \\ & - \tfrac{1}{L}\langle D_x - D_y, \nabla\phi_x - \nabla\phi_y\rangle \end{aligned} \tag{6.31}$$

Function ϕ is convex with Lipschitz gradients, therefore [77]:

$$\langle \nabla\phi(x) - \nabla\phi(y), x - y\rangle \geq \frac{1}{L}\|\nabla\phi(x) - \nabla\phi(y)\|^2 \tag{6.32}$$

Substituting (6.32) into (6.31) and using triangle inequality yields:

$$\begin{aligned} \langle D_x - D_y, x - y\rangle \geq \; & \|D_x - D_y\|^2 + \tfrac{1}{L^2}\|\nabla\phi_x - \nabla\phi_y\|^2 \\ & - \tfrac{1}{L}\|\nabla\phi_x - \nabla\phi_y\|\|D_x - D_y\| \end{aligned}$$

The above expression has the form $a^2 + b^2 - ab$, that satisfies:

$$a^2 + b^2 - ab = a^2 + \frac{1}{4}b^2 - ab + \frac{3}{4}b^2 = \left(a - \frac{1}{2}b\right)^2 + \frac{3}{4}b^2 \geq \frac{3}{4}b^2$$

where $a = \frac{1}{L}\|\nabla\phi_x - \nabla\phi_y\|, b = \|D_x - D_y\|$. We obtain (6.30):

$$\langle D_x - D_y, x - y\rangle \geq \frac{3}{4}\|D_x - D_y\|^2$$

∎

We establish the relationships between (6.23) and (6.24) in Lemma 6.4.6 and subsequently between (6.24) and the right hand of (6.21) in Lemma 6.4.7.

Lemma 6.4.6 *Consider two coarse corrections generated by taking m iterations from the coarse* *level at $x_{h,k}$ and $x_{h,*}$:*

$$\begin{aligned} d_{h,k} &= I_H^h e_{H,m} = I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i} \\ d_{h,*} &= I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i}^* = 0 \quad (\textit{from corollary}\ 6.4.4) \end{aligned} \tag{6.33}$$

then the following inequality holds:

$$\langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*} \rangle \geq \frac{1+2m}{4m\omega_2^2} \| d_{h,k} - d_{h,*} \|^2 \tag{6.34}$$

Proof From corollary 6.4.4, we know $D_{H,i}^* = 0, \forall i$, therefore:

$$\begin{aligned} \langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*} \rangle &= \left\langle x_{h,k} - x_{h,*}, I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i} - I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i}^* \right\rangle \\ &= \left\langle x_{H,0} - x_{H,0}^*, \sum_{i=0}^{m-1} s_{H,i} (D_{H,i} - D_{H,0}^*) \right\rangle \end{aligned} \tag{6.35}$$

Consider the i^{th} term of (6.35):

$$\begin{aligned} & s_{H,i} \left\langle x_{H,0} - x_{H,0}^*, D_{H,i} - D_{H,0}^* \right\rangle \\ =& s_{H,i} \langle x_{H,0} - x_{H,i} + x_{H,i} - x_{H,0}^*, D_{H,i} - D_{H,0}^* \rangle \\ \geq& s_{H,i} \langle x_{H,0} - x_{H,i}, D_{H,i} \rangle + \frac{3}{4} s_{H,i} \| D_{H,i} - D_{H,0}^* \|^2 \quad (\text{lemma 6.4.5 and } D_{H,0}^* = 0) \\ =& \langle x_{H,0} - x_{H,i}, x_{H,i} - x_{H,i+1} \rangle + \frac{3}{4 s_{H,i}} \| x_{H,i} - x_{H,i+1} \|^2 \quad \left(\text{as } D_{H,i} = \frac{x_{H,i} - x_{H,i+1}}{s_{H,i}} \right) \\ \geq& \langle x_{H,0} - x_{H,i}, x_{H,i} - x_{H,i+1} \rangle + \frac{3}{4} \| x_{H,i} - x_{H,i+1} \|^2 \quad (\text{as } s_{H,i} \in (0,1]) \end{aligned}$$

Substituting the above inequality in (6.35) yields:

$$\begin{aligned} & \langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*} \rangle \\ \geq & \sum_{i=0}^{m-1} \langle x_{H,0} - x_{H,i}, x_{H,i} - x_{H,i+1} \rangle + \frac{3}{4} \| x_{H,i} - x_{H,i+1} \|^2 \\ = & \underbrace{3/4 \| x_{H,0} - x_{H,1} \|^2 + \langle x_{H,0} - x_{H,1}, x_{H,1} - x_{H,2} \rangle + 3/4 \| x_{H,1} - x_{H,2} \|^2}_{\Delta} \\ & + \sum_{i=2}^{m-1} \langle x_{H,0} - x_{H,i}, x_{H,i} - x_{H,i+1} \rangle + 3/4 \| x_{H,i} - x_{H,i+1} \|^2 \end{aligned} \tag{6.36}$$

The quantity Δ has the form:

$$\frac{3}{4}a^2 + ab + \frac{3}{4}b^2 = \frac{1}{2}(a+b)^2 + \frac{1}{4}a^2 + \frac{1}{4}b^2 \tag{6.37}$$

Utilising (6.37) in (6.36) we obtain:

$$\begin{aligned}
& \langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*} \rangle \\
= \ & \frac{1}{4}\|x_{H,0} - x_{H,1}\|^2 + \frac{1}{4}\|x_{H,1} - x_{H,2}\|^2 \\
& + \frac{1}{2}\|x_{H,0} - x_{H,2}\|^2 + \langle x_{H,0} - x_{H,2}, x_{H,2} - x_{H,3} \rangle + \frac{1}{2}\|x_{H,2} - x_{H,3}\|^2 + \frac{1}{4}\|x_{H,2} - x_{H,3}\|^2 \\
& + \sum_{i=3}^{m-1} \langle x_{H,0} - x_{H,i}, x_{H,i} - x_{H,i+1} \rangle + \frac{1}{2}\|x_{H,i} - x_{H,i+1}\|^2 + \frac{1}{4}\|x_{H,i} - x_{H,i+1}\|^2 \\
= \ & \frac{1}{4}\sum_{i=0}^{m-1} \|x_{H,i} - x_{H,i+1}\|^2 + \frac{1}{2}\|x_{H,0} - x_{H,m}\|^2 \\
\geq \ & \frac{1}{4m}\left(\sum_{i=0}^{m-1} \|x_{H,i} - x_{H,i+1}\|\right)^2 + \frac{1}{2}\|x_{H,0} - x_{H,m}\|^2 \quad \text{(Cauchy-Schwarz)} \\
\geq \ & \frac{1}{4m}\|x_{H,0} - x_{H,m}\|^2 + \frac{1}{2}\|x_{H,0} - x_{H,m}\|^2 \quad \text{(triangle-inequality)} \\
= \ & \frac{1+2m}{4m}\|e_{H,m}\|^2 \\
\geq \ & \frac{1+2m}{4m\omega_2^2}\left\|I_H^h e_{H,m}\right\|^2 \quad \text{(assumption 6.4.2)} \\
= \ & \frac{1+2m}{4m\omega_2^2}\|d_{h,k} - d_{h,*}\|^2
\end{aligned}$$

∎

Having shown the relationship between the cross term (6.23) and the norm of the coarse corrections (6.24), we now establish the connection between (6.24) and the distance of $x_{h,k}$ to optimal solution (the right side of (6.21)).

Lemma 6.4.7 *Consider the coarse correction terms defined by* (6.33). *Then we have the following inequality:*

$$\|d_{h,k} - d_{h,*}\|^2 \leq \frac{16}{9}m^2\omega_1^2\omega_2^2 s_{H,0}^2\|x_{h,k} - x_{h,*}\|^2 \tag{6.38}$$

where ω_1, ω_2 are defined in assumption 6.4.2.

Proof At the coarse level H, assume the contraction algorithm is utilised on the level H.

Therefore, we have:

$$\|x_{H,k+1} - x_{H,k}\| \leq \|x_{H,k} - x_{H,k-1}\|$$

or,

$$s_{H,k}\|D_{H,k}\| \leq s_{H,k-1}\|D(x_{H,k-1})\| \tag{6.39}$$

Now, we have:

$$\begin{aligned}
& \|d_{h,k} - d_{h,*}\|^2 \\
= \; & \left\| I_H^h \sum_{i=0}^{m-1} s_{H,i} D_{H,i} - 0 \right\|^2 \\
\leq \; & \omega_2^2 \left\| \sum_{i=0}^{m-1} s_{H,i} D_{H,i} \right\|^2 \quad \text{(by assumption 6.4.2)} \\
\leq \; & \omega_2^2 \left(\sum_{i=0}^{m-1} s_{H,i} \|D_{H,i}\| \right)^2 \quad \text{(by triangle inequality)} \\
\leq \; & m^2 \omega_2^2 s_{H,0}^2 \|D_{H,0}\|^2 \quad \text{(using (6.39))} \\
= \; & m^2 \omega_2^2 s_{H,0}^2 \|D_{H,0} - D_{H,0}^*\|^2 \quad \text{(as } D_{H,0}^* = 0\text{)} \\
= \; & m^2 \omega_2^2 s_{H,0}^2 \|I_h^H D_{h,k} - I_h^H D_{h,*}\|^2 \quad \text{(using (6.25))} \\
\leq \; & m^2 \omega_1^2 \omega_2^2 s_{H,0}^2 \|D_{h,k} - D_{h,*}\|^2 \quad \text{(by assumption 6.4.2)} \\
\leq \; & \frac{4}{3} m^2 \omega_1^2 \omega_2^2 s_{H,0}^2 \langle D_{h,k} - D_{h,*}, x_{h,k} - x_{h,*} \rangle \quad \text{(by lemma 6.4.5)} \\
\leq \; & \frac{16}{9} m^2 \omega_1^2 \omega_2^2 s_{H,0}^2 \|x_{h,k} - x_{h,*}\|^2
\end{aligned}$$

■

The contraction property of the coarse correction update at the step $x_{h,k}$ is established in the following theorem.

Theorem 6.4.8 (Contraction for coarse correction update) *For a coarse correction update at iteration $x_{h,k}$, suppose either the assumption 6.4.2 is satisfied, i.e. $\omega_1 \geq 1$ and $\omega_2 \leq 1$, or the number of coarse iterations m is sufficiently large, then,*

$$\|x_{h,k+1} - x_{h,*}\|^2 \leq \sigma(s, \tau)\|x_{h,k} - x_{h,*}\|^2 \tag{6.40}$$

where $\sigma(\tau, s) = 2 + \Delta(\tau)s^2$. There always exists $\tau > 0$ such that,

$$\Delta(\tau) < -1$$

and,

$$\frac{1}{\sqrt{-\Delta(\tau)}} \leq s \leq \min\left\{\frac{2}{\sqrt{-\Delta(\tau)}}, 1\right\} \tag{6.41}$$

Consequently, we have,

$$\sigma(\tau, s) < 1$$

.

Proof Consider the norm

$$\begin{aligned}
& \|x_{h,k+1} - x_{h,*}\|^2 \\
= \;& \|[x_{h,k} - s(x_{h,k} - \text{prox}_h(x_{h,k} - \tau d_{h,k}))] - [x_{h,*} - s(x_{h,*} - \text{prox}_h(x_{h,*} - \tau d_{h,*}))]\|^2 \\
= \;& \|(1-s)(x_{h,k} - x_{h,*}) + s[\text{prox}_h(x_{h,k} - \tau d_{h,k}) - \text{prox}_h(x_{h,*} - \tau d_{h,*})]\|^2 \\
\leq \;& 2(1-s)^2\|x_{h,k} - x_{h,*}\|^2 + 2s^2\|\text{prox}_h(x_{h,k} - \tau d_{h,k}) - \text{prox}_h(x_{h,*} - \tau d_{h,*})\|^2 \quad \text{(by Cauchy-Schwarz)} \\
\leq \;& 2(1-s)^2\|x_{h,k} - x_{h,*}\|^2 + 2s^2\|(x_{h,k} - \tau d_{h,k}) - (x_{h,*} - \tau d_{h,*})\|^2 \quad \text{(by nonexpansive)} \\
= \;& (4s^2 - 4s + 2)\|x_{h,k} - x_{h,*}\|^2 + 2s^2(\tau^2\|d_{h,k} - d_{h,*}\|^2 - 2\tau\langle x_{h,k} - x_{h,*}, d_{h,k} - d_{h,*}\rangle). \qquad (6.42)
\end{aligned}$$

As $s \in (0, 1] \Rightarrow 4s^2 - 4s \leq 0$, and from lemma 6.4.6, we have:

$$(6.42) \leq 2\|x_{h,k} - x_{h,*}\|^2 + \frac{4m\omega_2^2\tau^2 - 2\tau(1+2m)}{2m\omega_2^2}s^2\|d_{h,k} - d_{h,*}\|^2 \tag{6.43}$$

From Lemma 6.4.7, we have:

$$(6.43) \leq \left(2 + \underbrace{\frac{8}{9}m\omega_1^2 s_{H,0}^2(4m\omega_2^2\tau^2 - 2\tau(1+2m))}_{\Delta(\tau)}.s^2\right)\|x_{h,k} - x_{h,*}\|^2$$

The contraction property requires,

$$0 < 2 + \Delta(\tau)s^2 < 1 \Rightarrow -2 < \Delta(\tau)s^2 < -1.$$

As $s \in (0,1]$, it is essential that,

$$\Delta(\tau) < -1$$

Therefore, we need to find a τ that satisfies,

$$A^2\tau^2 - B\tau + 1 < 0 \tag{6.44}$$

where

$$\begin{aligned} A^2 &= \frac{32}{9}m^2\omega_1^2\omega_2^2 s_{H,0}^2 \Rightarrow 2A = \frac{8\sqrt{2}}{3}m\omega_1\omega_2 s_{H,0} \\ B &= \frac{16}{9}m(1+2m)\omega_1^2 s_{H,0}^2. \end{aligned}$$

Inequality (6.44) can be written as,

$$(A\tau - 1)^2 - (B - 2A) < 0$$

The above inequality is always satisfied for $B > 2A$. For example, we can set $\tau = 1/A$ and if the assumption 6.4.2 is satisfied, i.e. $\omega_1 \geq 1$ and $\omega_2 \leq 1$, then $2A$ is always less than B. On the other hand, when the assumption 6.4.2 is not satisfied, but the number of coarse iterations m is sufficiently large, then B is also greater than $2A$.

Once τ is defined such that $\Delta(\tau) < -1$, we can deduce:

$$\frac{1}{\sqrt{-\Delta(\tau)}} \leq s \leq \min\left\{\frac{2}{\sqrt{-\Delta(\tau)}}, 1\right\}.$$

∎

Finally, at any iteration of the fine level, regardless of updating by gradient proximal step (6.19), or coarse correction step (6.18), we always have a contraction property due to Lemma 6.4.1 and Theorem 6.4.8. Let the worst contraction modulus be:

$$\sigma = \sqrt{\max\{\sigma_{ISTA}, \sigma(\tau, s)\}} \tag{6.45}$$

Obviously, $\sigma \in (0,1)$, and we can summarise the linear convergence rate for MISTA in the following corollary,

Corollary 6.4.9 *The sequence generated by algorithm 6.1 converges linearly to the optinum $x_{h,*}$:*

$$\|x_{h,k+1} - x_{h,*}\| \leq \sigma \|x_{h,k} - x_{h,*}\| \tag{6.46}$$

where $\sigma \in (0, 1)$ is a contraction modulus defined in (6.45).

6.5 Numerical examples

In this section we illustrate the numerical performance of the algorithm by the image restoration problem. We report results on CPU time and convergence of objective function on a large set of images. All images size are 1024 × 1024, the large image size requires more computational resource for a fine model, and thus we can easily observe the advantage of utilising our multilevel methods. We implemented the ISTA and FISTA algorithms from [6] using the same parameter settings. We call an iteration an ISTA or FISTA step if the incumbent solution is updated using ISTA or FISTA respectively. We tested two variants of the proposed MISTA algorithm. We refer to the first variant as MISTA-I (Multilevel Iterative Thresholding Algorithm-I). MISTA-I employs the updates of algorithm 6.1. We refer to the second variant as MISTA-F (Multilevel Iterative Thresholding Algorithm-F). MISTA-F employs the updates of algorithm 6.2. Note that the theory developed in this chapter does not cover the case where a FISTA step is performed. We consider MISTA-F in this case because the performance of the algorithm in this case is very promising and and we are currently investigating this line of research. The step-size strategy for MISTA-I and MISTA-F was selected according the backtracking line search for projection algorithm presented in Algorithm 6.3 [91]. The condition to use the coarse model to compute a search direction is shown in Corollary 6.4.4, with $\kappa = 0.49$ and $\eta = 1$. The lowest resolution allowed for the coarse construction is 256 × 256. All algorithms were implemented in MATLAB and run on a standard desktop PC.

Consider the image restoration problem described in Section 6.3.4. The fine model is defined as the composite function:

$$\min_{x_h \in \Omega_h} \|A_h x_h - b_h\|^2 + \lambda_h \|W(x_h)\|$$

Algorithm 6.2: Fast Multilevel Iterative Shrinkage Thresholding Algorithm

```
if Condition to restrict current iterate $x_{h,k}$ to coarse model is satisfied then
    Set $x_{H,0} = I_h^H x_{h,k}$;
    Compute $m$ iterations of the coarse level
        $$x_{H,m} = x_{H,0} + \sum_{i=0}^{m} s_{H,i} D_{H,i}$$
    Set $d_{h,k} = I_H^h (x_{H,0} - x_{H,m})$;
    Find a suitable $\tau$ that satisfies (6.44), compute:
        $$x^+ = \text{prox}_h(x_{h,k} - \tau d_{h,k})$$
    Choose a step-size $s \in (0, 1]$ that satisfies (6.41) to update:
        $$x_{h,k+1} = x_{h,k} - s(x_{h,k} - x_h^+) \tag{6.47}$$
else
    Compute gradient mapping:
        $$D_{h,k} = y_{h,k} - \text{prox}_h(y_{h,k} - \frac{1}{L_h}\nabla f(y_{h,k}))$$
    Choose a step-size $s \in (0, 1]$ to update:
        $$x_{h,k+1} = y_{h,k} - sD_{h,k} \tag{6.48}$$
$$t_{k+1} = \left(1 + \sqrt{1 + 4t_k^2}\right)/2$$
$$y_{k+1} = x_{h,k+1} + (t_k - 1)/(t_{k+1})(x_{h,k+1} - x_{h,k})$$
```

Algorithm 6.3: Backtracking linesearch for MISTA-{I,F}

```
Set $\gamma \in (0,1), i = 1, c \in (0,1)$;
Compute a coarse correction: $\delta = x_{h,k} - \text{prox}_h(x_{h,k} - \tau d_{h,k})$ ;
Choose a step-size $s$ that satisfies (6.41);
repeat
    $s = s * \gamma^i$;
    $i = i + 1$;
until $f(x_{h,k} - s\delta) \leq f(x_{h,k}) - cs\langle \nabla f(x_{h,k}), \delta \rangle$;
```

while the coarse model is approximated by smoothing the l_1-norm:

$$\min_{x_H \in \Omega_H} \|A_H x_H - b_H\|^2 + \lambda_H \sum_{i \in x_H} \sqrt{W(x_H^i)^2 + \mu^2} - \mu$$

where $\mu = 0.2$ is a smoothing parameter, λ is the regulariser parameter and initially set to $1e-5$. As the dimension gets lower, the coarse problem is smoother, therefore the regularising levels should be reduced, e.g. $\lambda_H = \lambda_h/2$. In addition, b_h and x_h are the vectorised forms of the input corrupted image B_h and variable X_h respectively. A_h is the blurring operator based on the point spread function (PSF) and reflexive boundary condition. Utilising efficient implementation provided in the HNO package [41], we can rewrite the huge matrix computation $A_h x_h - b_h$ in a reduced form:

$$A_h^c X_h A_h^r{}' - B_h$$

where A_h^c, A_h^r are the row/column blurring operators and $A_h = A_h^r \otimes A_h^c$. The information transfer between levels is done via a simple linear interpolation technique to group 4 finely discretised pixels into 1 coarse pixel, as similar to the one mentioned in Assumption 6.4.2:

$$x_{H,0} = I_h^H x_{h,k} \quad , \quad b_H = I_h^H b_h$$

The standard matrix restriction $A_H = I_h^H A_h I_h^H{}'$ is not performed explicitly as we never need to store the huge matrix A_h. Instead, only column and row operators A_h^c, A_h^r are stored in the computer memory. As a decomposition of the restriction operator is available for our problem, in particular $I_h^H = R_1 \otimes R_2$, we can obtain the coarse blurring matrix by:

$$A_H = A_H^r \otimes A_H^c$$

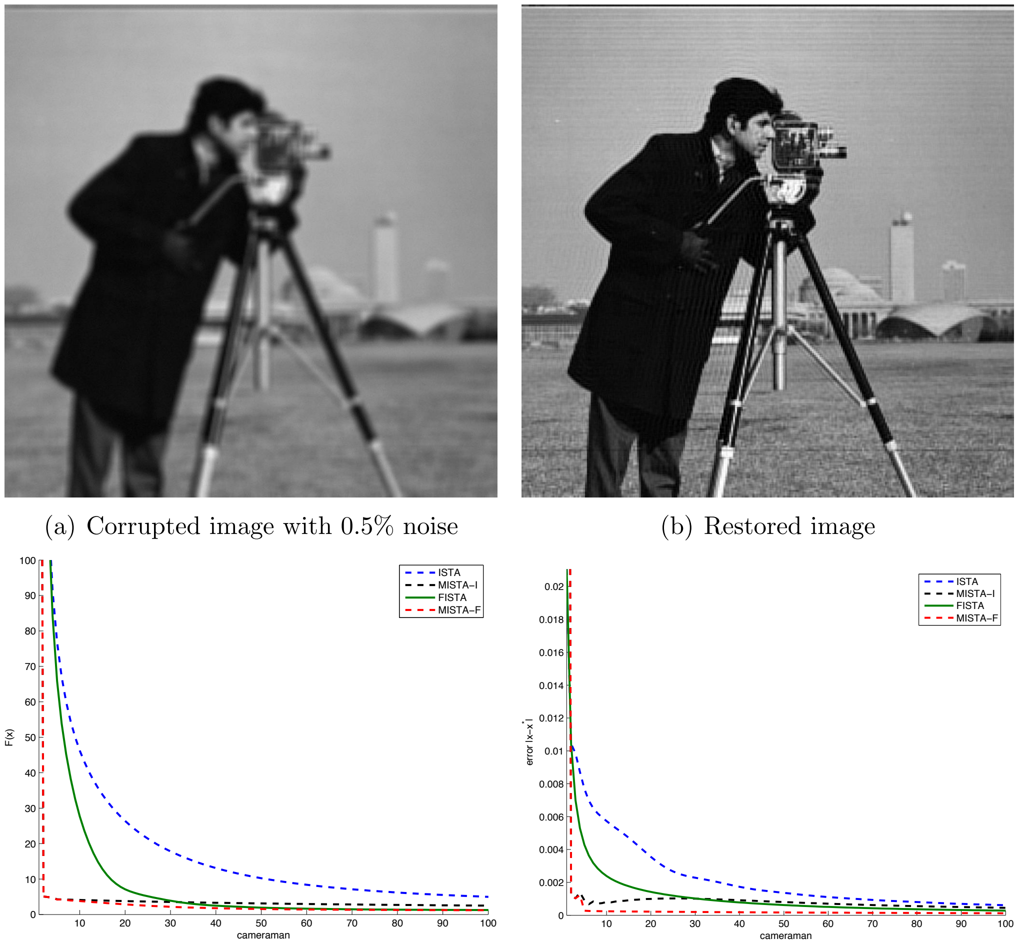
where $A_H^c = R_2 A_h^c R_1'$ and $A_H^r = R_1 A_h^r R_2'$.

We compare the performance of our methods with FISTA/ISTA using a large set of corrupted images (blurred with 0.5% additive noise) including the famous cameraman image, see Figure 6.4(a), as a standard benchmark. The restored image is shown in Figure 6.4(b). In Figure 6.4(c) we compare the four algorithms in terms of the progress they make in function value reduction. In this case we see that both versions of MISTA clearly outperform ISTA. This result is not surprising since MISTA is a more specialized algorithm with the same convergence properties. However, what is surprising is that both MISTAs still outperform the theoretically superior FISTA algorithm. Clearly, MISTA-F is always the best algorithm, while MISTA-I outperforms FISTA in early iterations and is comparable in latter iterations. When we compare the distance to the optimal vector in Figure 6.4(d) we again find both versions of MISTA outperform ISTA. MISTA-F finds optimal solution after only 5 iterations while other still not converges after 100 iterations. However in this case FISTA and MISTA-I are comparable. We also observe that there is a clear advantage of MISTA-I in early iterations. This may be important in applications where an approximate solution is required but due to time limitations the number of iterations has to be kept low. We performed similar experiments in a number of images and from these we have come to the conclusion that both MISTA variants outperform ISTA. We found that there is a clear advantage of using MISTA in early iterations.

Figure 6.4 gives some idea of the performance of the algorithm but of course what matters most is the CPU time required to compute a solution. In order to shed more light into this issue, we discuss the performance of the algorithms when we tested them on a benchmark suite of images. Two experiments were performed on a set of 6 images as reported in this section. The first experiment takes input blurred images with 0.5% additive Gussian noise and the second experiment uses 1% additive noise. All these images and the MATLAB package are also available from the project website. In order to make our results easier to read we divided the CPU time required to find an optimum solution that is within 2% of the optimality conditions with the CPU time taken by ISTA,

$$\text{Improvement rate} = \text{ISTA CPU time} \;/\; \text{other method CPU time}$$

ISTA was the slower algorithm from the ones we tried so we used it as a baseline. Since the plot shows improvement over ISTA the higher the value the better the algorithm. We expect the experiment with 1% additive noise is more difficult to solve than 0.5% noise, as the corrupted



(a) Corrupted image with 0.5% noise

(b) Restored image

(c) Function value comparison

(d) Error comparison

Figure 6.4: *(a) Corrupted cameraman image used as the input vector b in* (6.20)*, (b) Restored image, (c) Comparison of the three algorithms in terms of function value. Both versions of MISTA outperform ISTA. When compared to FISTA there are clear advantages in early iterations from using MISTA (d) Distance from optimum vector $x^\star$. The optimal solution $x^\star$ was computed with* 10^4 *iterations of the FISTA algorithm. Note that in terms of distance to the optimum MISTA-F clearly outperforms the other algorithms and converges in essentially 5 iterations, while others are not converged even after 100 iterations.*

image problem is more ill-conditioned with greater noise. Figure 6.5 shows the performance of blurred images with 0.5% noise. We can see that both versions of MISTA outperform ISTA/FISTA significantly. MISTA-I is at least 4.5 times faster than ISTA, and 1.5 times faster than FISTA. MISTA-F is at least 6 times faster than ISTA, and twice as fast as FISTA. However, on average, both variants of MISTA is 4 times faster than FISTA and 10 times faster than ISTA. In figure 6.6, we see even greater improvement of MISTA-I/F over ISTA/FISTA. This is expected since the problem is more ill-conditioned (with 1% noise as opposed to 0.5% noise in Figure 6.5), and the fine level requires more iterations to converge. As the results,

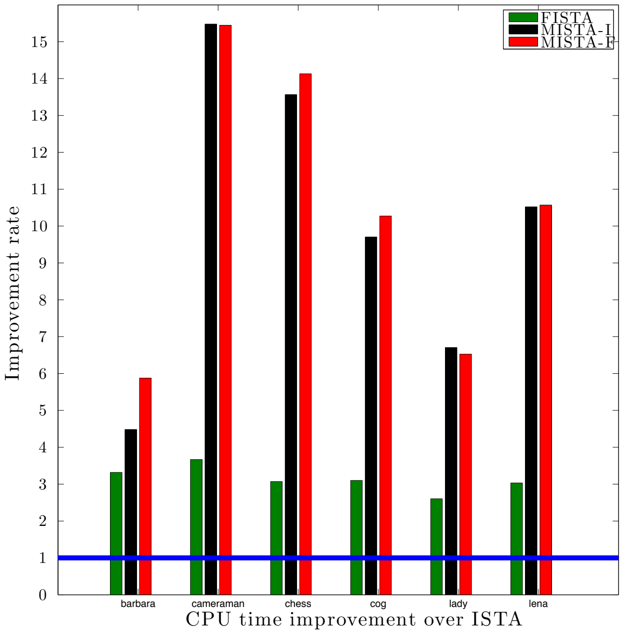


Figure 6.5: *Comparison in terms of improvement of FISTA and MISTA-I/F when compared to ISTA. A higher value means bigger improvement in terms of CPU time required to find a solution within 2% of the optimum. Images are blurred with 0.5% noise.*

CPU time of ISTA/FISTA at the fine level increase. On the other hand, the convergence of MISTA-I/F depends on how well the coarse correction impacts on the fine level and the CPU time of MISTA depend mostly on solving the coarse model. And, the CPU time of MISTA-I/F at the coarse level are only marginally increased because the greater ”ill-conditioned” of the fine problem (with 1% noise) has less effects on the coarse model.

6.6 Conclusions

In this section, we develop a novel multilevel optimisation framework that can handle problems with nondifferentiable objective function and simple constraints. The key components of the framework are the good coarse approximations of the fine model and the local equivalence between levels based on first order coherence. In this thesis, we consider the most basic prolongation and restriction operators in approximating the coarse model. The literature on the construction of these operators is quite large and there exists more advanced operators that adapt to the problem data and current solution (e.g. bootstrap AMG [18]). We expect that the numerical performance of the algorithm can be improved if these advanced techniques are

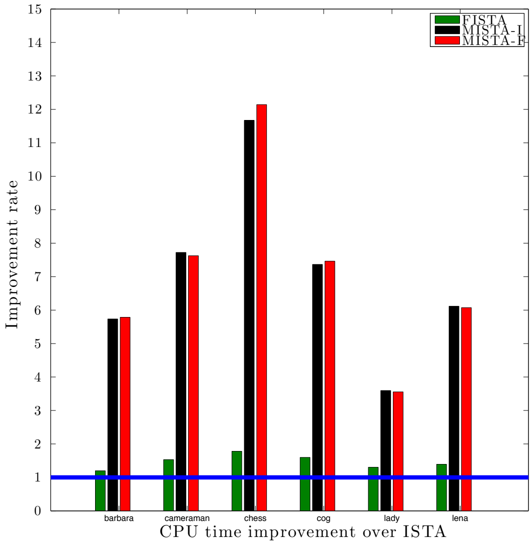
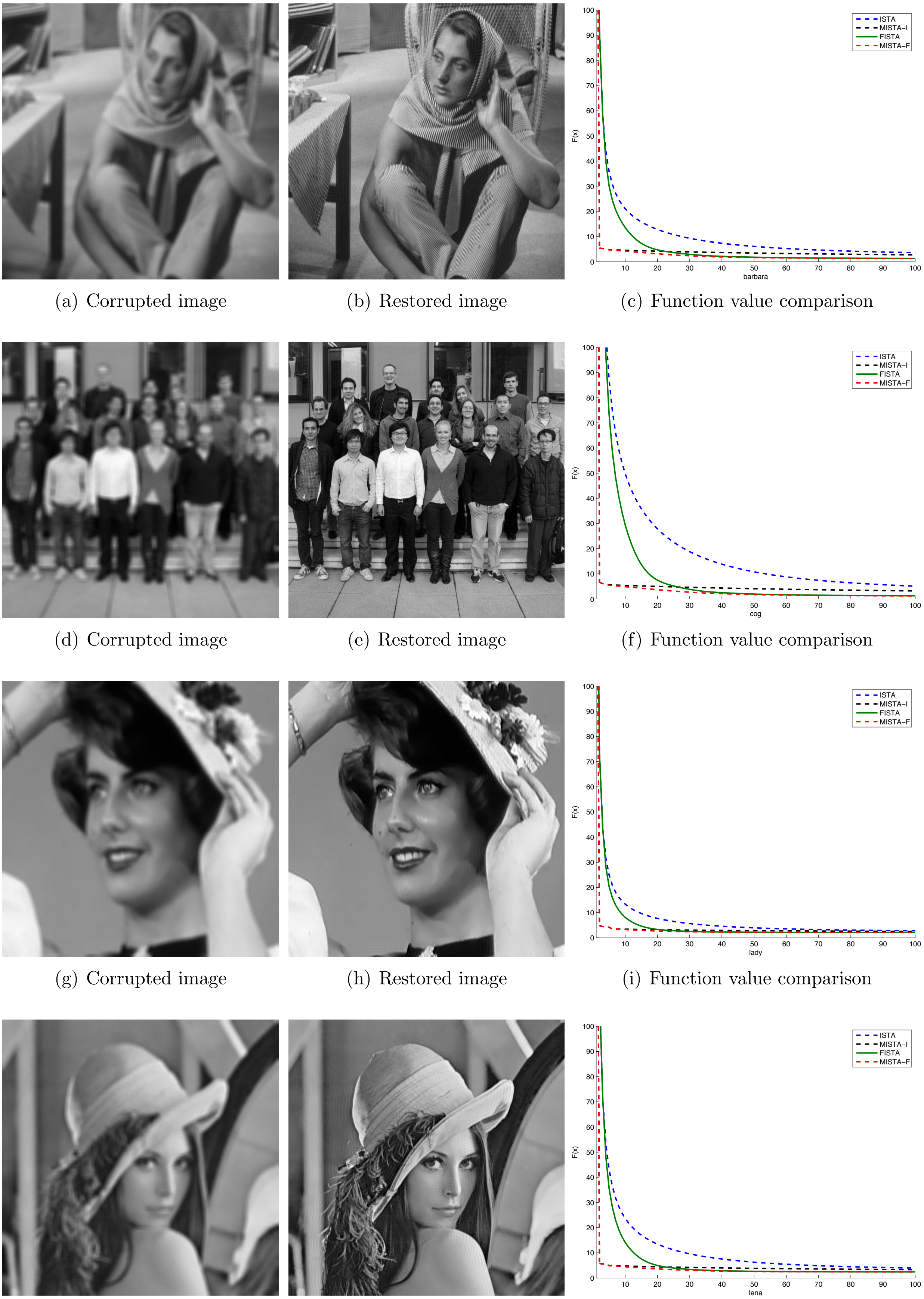


Figure 6.6: *Comparison in terms of improvement of FISTA and MISTA-I/F when compared to ISTA. A higher value means bigger improvement in terms of CPU time required to find a solution within 2% of the optimum. Images are blurred with 1% noise.*

used instead of the naive approach proposed here.



(a) Corrupted image (b) Restored image (c) Function value comparison

(d) Corrupted image (e) Restored image (f) Function value comparison

(g) Corrupted image (h) Restored image (i) Function value comparison

(j) Corrupted image (k) Restored image (l) Function value comparison

Chapter 7

Conclusions

This thesis considers solutions to three problems in image processing. All algorithms are considered for various common type of relaxations in image processing. We report progress in each case, based on experimental and theoretical results. In this chapter, we summarise the achievements and propose future directions of research.

7.1 Summary of thesis

In Chapters 2 and 3, we solve the image registration problem by nonlinear programming. Image registration is an illed-posed problem and suffers from many undesired local optimal solutions. A common approach to reduce the non-uniqueness of the problem is to add suitable regularisers and constraints. In this work, various regularisers and rigidity constraints are incorporated into the registration problem. Constrained image registration is an important problem in medical imaging, where the addition of the constraints can improve the accuracy and feasibility of registration in clinical applications. Rigid constraints ensure the rigid movement of rigid objects (such as bone) in the images. The constraints are formulated by considering the first and second order derivatives of the transformation vector at every rigid pixel. There are two issues to consider in this problem: discretisation and optimisation. As input images are infinite dimensional, discretisation is required to formulate a finite dimensional optimisation problem. Due to the complexity of the constraints, staggered grid discretisation is used to address the first issue. We develop and adapt the SQP algorithm with a dimensionality reduction technique to solve the finite constrained optimisation problem. Experimental results demonstrate significant improvement over an unconstrained gradient method (FLIRT)[70] to solve the same constrained registration problem, in terms of CPU performance and retaining feasibility. We believe that the behaviour of the transformation grid (produced by our algorithm) on rigid objects is more reliable for medical analysis.

Image registration is a typical image processing problem that requires relaxations to remove undesirable solutions. Apart from the introduction of regularisers and constraints that reduces the non-uniqueness of the suboptimal solutions, convex relaxation is also a common approach. The relaxation suffers from the loss of accuracy but benefits from the availability of efficient convex optimisation solvers. One popular relaxation framework in image processing is to construct a Markov Random Field (MRF) model. MRF has been successfully used for applications in image and signal processing, machine learning and artificial intelligence. The original MRF is a nonconvex integer linear programming problem. There exist many studies for solving the MRF approximately by dynamic programming, combinatorial optimisation, or convex relaxation. In this thesis, we consider the dual of the LP relaxation of MRF (LP-MRF). The dual LP-MRF is a large dimensional, convex-nondifferentiable optimisation problem with simple linear constraints. We propose a nonlinear weighted projection algorithm based on the mirror descent approach to solve the dual LP-MRF. We sharpen the convergence rate and show promising experimental results via synthetic and an image segmentation problem. Furthermore, we believe that although our method does not have an accelerated convergence rate as the smoothing approach [92], its computational cost per iteration is very low. As the result, the proposed method provides a reasonable alternative to the state-of the art methods for solving MRF models.

The above two methods (i.e. the SQP and mirror descent algorithms) solve two typical types of relaxations to image processing problems. They take into account the image structure at a certain level of discretisation. Both the SQP algorithm, and mirror descent (with weighted projection) cease to be of practical use when considering applications to very high dimensional problems. To overcome this difficulty, we introduce a novel approach employing different levels of discretisation for the optimisation solver. The new algorithms are based on the gradient proximal method (ISTA/FISTA) that can handle convex problems with simple constraints and simple nonsmooth regularisers. In order to reduce the computational effort needed for large problems, we propose to use low-cost steps, generated using a solution to a coarse discretised version of the problem. These replace some of inefficient gradient updates needed by the finely (or, more accurately) discretised version of the original problem. The global convergence (at the finest level) is guaranteed via the contraction property on a distance to the optimal solution. The convergence analysis is based mainly on the first order coherence of the problem. First order coherence establishes the local equivalence between the gradient mapping of the fine model and the gradient mapping of the coarse model. Experimental results show excellent performance of our methods compared to the state-of-the-art gradient proximal methods (ISTA/FISTA). We believe that our method provides a broad framework for solving large image processing problems. The experimental results suggest that it can be applicable for other real computer vision problems. Therefore, there is considerable scope for further development of specialised algorithms utilising the multilevel framework developed in this thesis.

7.2 Future work

The methods proposed in Chapters 2 and 3 can be directly extended to 3D images. In addition, some studies of constrained *parametric* registration framework [88, 99] consider similar constraints as in this thesis. All these adopt the *unconstrained penalty method* for solving the parametric constrained registration problem. An exciting direction of future work is to employ the SQP algorithm with the dimensionality reduction technique (developed in this thesis) to solve the constrained registration problem in a parametric framework[90, 94].

The dual formulation of MRF problem in Chapters 4 and 5 only considers the tree structures for the simple MRF subproblems (the slave). However, our proposed method is sufficiently general for employing alternative decompositions such as edges decomposition and loop decomposition [58]. Extended experiments can be implemented to compare the performance of various decompositions techniques applied to computer vision problems. Another possible line of enquiry addresses the issue that the significant improvement of the proposed method results from the weighted entropy projection. However, the improvement is no longer evident when the method finds the optimal point in the bounded simplex sets. At this point, the algorithm switches to the weighted Euclidean projection. One possible future work is to dynamically adjust the bound of the simplex sets based on the primal-dual gap. Theoretical justification is the subject of ongoing work.

Chapter 6 provides a very general framework for multilevel optimisation for image processing. There is considerable scope for future research in defining suitable coarse models, information transformation between levels, or efficient methods for solving the coarse model. In this thesis, we consider the most basic restriction/prolongation operator that groups four pixels at the fine level into one corresponding coarse pixel. There are more advanced techniques to define operators that adapt to the problem and current solution, and it can reasonably be expected that these will lead to greater improvement. Finally, the convergence analysis for the multilevel framework using the Newton algorithm and accelerated proximal gradient (FISTA) are still open issues on which we are working.

In conclusion, several ideas presented in this thesis are potentially applicable to real-world problems. Ongoing work continues to extend some theoretical results, while, in collaboration with computer vision experts, applies our ideas to improve the performance of large scale practical computer vision problems.

Bibliography

[1] Barker, S. A. *Image Segmentation using Markov Random Field Models.* PhD thesis, Wolfson College, University of Cambridge, 1998.

[2] Batra, D., Gallagher, A., Parikh, D., and Chen, T. Beyond trees: Mrf inference via outer-planar decomposition. *Proc. of CPVR 2010* (2010).

[3] Beck, A., and Sabach, S. A first order method for finding minimal norm-like solutions of convex optimization problems. *Mathematical Programming* (2013).

[4] Beck, A., and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters 31* (2003).

[5] Beck, A., and Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Trans. Img. Proc. 18*, 11 (2009).

[6] Beck, A., and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 1 (2009), 183–202.

[7] Beck, A., and Teboulle, M. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization 22* (2012).

[8] Ben-Tal, A., Margalit, T., and Nemirovski, A. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal on Optimization 12* (2001).

[9] Bertsekas, D. P. *Nonlinear Programming.* Optimization and Computation Series. Athena Scientific, 1999.

[10] Besag, J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society* (1986).

[11] Bishop, C. M. *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer-Verlag, 2006.

[12] Borzì, A., and Schulz, V. Multigrid methods for pde optimization. *SIAM review 51*, 2 (2009), 361–395.

[13] Borz, A. On the convergence of the mg/opt method. *Proc. in Applied Mathematics and Mechanics* (2005).

[14] Boyd, S., and Vandenberghe, L. *Convex Optimization.* Cambridge University Press, 2004.

[15] Boykov, Y., and Jolly, M. P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *Proc. of ICCV '01* (2001).

[16] Boykov, Y., and Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis and Machine Intelligence 26* (2004).

[17] Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence 23* (2001).

[18] Brandt, A., Brannick, J., Kahl, K., and Livshits, I. Bootstrap amg. *SIAM Journal on Scientific Computing 33* (2011).

[19] Bredies, K., and Lorenz, D. Linear convergence of iterative soft-thresholding. *Journal of Fourier Analysis and Applications 14* (2008).

[20] Briggs, W. L., Henson, V. E., and McCormick, S. F. *A multigrid tutorial*, second ed. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[21] Broit, C. *Optimal registration of deformed images.* PhD thesis, Computer and Information Science, University of Pensylvania, USA, 1981.

[22] Chambolle, A., Devore, R., Lee, N., and Lucier, B. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Processing 7* (1998).

[23] Chekuri, C., Khanna, S., Naor, J., and Zosin, L. Approximation algorithms for the metric labelling problem via a new linear programming formulation. *Proc. of ACM-SIAM Symposium on Discrete Algorithms* (2001).

[24] Chen, G., and Teboulle, M. Convergence analysis of a proximal like minimization algorithm using bregman functions. *SIAM Journal on Optimization 3* (1993).

[25] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l1-ball for learning in high dimensions. *Proc. of ICML '08* (2008).

[26] Facciolo, G., Almansa, A., Aujol, J.-F., and Caselles, V. Irregular to regular sampling, denoising and deconvolution. *SIAM Journal on Multiscale Modeling and Simulation 7* (2009).

[27] Fischer, B., and Modersitzki, J. Curvature based image registration. *Journal of Mathematical Imaging and Vision 18* (2003).

[28] Freeman, W., Pasztor, E., and Carmichael, O. Learning low-level vision. *International Journal of Computer Vision 40* (2000).

[29] Frey, B., and MacKay, D. A revolution: Belief propagation in graphs with cycles. *Advances in Neural Information Processing Systems* (1997).

[30] Globerson, A., and Jaakkola, T. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. *Proc. of NIPS '08* (2008).

[31] Glocker, B., Komodakis, N., Tziritas, G., Navab, N., and Paragios, N. Dense image registration through mrfs and efficient linear programming. *Medical Image Analysis* (2008).

[32] Glocker, B. M. *Random Fields for Image Registration.* PhD thesis, Technischen University Munich, 2010.

[33] Gratton, S., Mouffe, M., Sartenaer, A., Toint, P. L., and Tomanos, D. Numerical experience with a recursive trust-region method for multilevel nonlinear bound-constrained optimization. *Optimization Methods & Software 25*, 3 (2010), 359–386.

[34] Gratton, S., Mouffe, M., Toint, P. L., and Weber-Mendonça, M. A recursive-trust-region method for bound-constrained nonlinear optimization. *IMA Journal of Numerical Analysis 28*, 4 (2008), 827–861.

[35] Gratton, S., Sartenaer, A., and Toint, P. L. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization 19* (2008).

[36] Haber, E., and Ascher, U. Fast finite volume simulation of 3d electromagnetic problems with highly discontinuous coefficients. *SIAM Journal of Scientific Computing 22* (2001).

[37] Haber, E., and Modersitzki, J. Numerical methods for volume preserving image registration. *Inverse Problems 20* (2004).

[38] Haber, E., and Modersitzki, J. A multilevel method for image registration. *SIAM Journal on Scientific Computing* (2006).

[39] Hajnal, J. V., Hill, D. L., and Hawkes, D. J. *Medical Image Registration (Biomedical Engineering).* CRC Press, 2001.

[40] Hammersley, J., and Clifford, P. Markov fields on finite graphs and lattices. *Unpublished Manuscripts* (1971).

[41] Hansen, P. C., Nagy, J. G., and O'Leary, D. P. *Deblurring Images: Matrices, Spectra, and Filtering.* Fundamentals of Algorithms. Siam Philadelphia, 2006.

[42] J., F. C. A. *Computational Techniques for Fluid Dynamics*, vol. 2. Berlin: Springer, 1988.

[43] Jancsary, J., and Matz, G. Convergent decomposition solvers for tree-reweighted free energies. *Proc. of AISTATS 2011* (2011).

[44] Jancsary, J., Matz, G., and Trost, H. An incremental subgradient algorithm for map estimation in graphical models. *Proc. of NIPS '10 Workshop on Optimization for Machine Learning* (2010).

[45] Johnson, J. K., Malioutov, D. M., and Willsky, A. S. Lagrangian relaxation for map estimation in graphical models. *Proc. of the 45th Allerton Conference on Communication, Control and Computing* (2007).

[46] Jojic, V., Gould, S., and Koller, D. Accelerated dual decomposition for map inference. *Proc. of ICML '10* (2010).

[47] Juditsky, A., Nazin, A., Tsybakov, A., and Vayatis, N. Recursive aggregation of estimators by the mirror descent algorithm with averaging. *Problems of Information Transmission* (2005).

[48] Juditsky, A., and Nemirovski, A. *First order methods for nonsmooth convex large-scale optimization, I: General purpose methods.* Optimization for Machine Learning. The MIT Press, 2012, ch. 5.

[49] Kelley, C. T. *Iterative methods for optimization*, vol. 18. Siam, 1999.

[50] Kervrann, C., and Boulanger, J. Unsupervised patch-based image regularization and representation. *Proc. ECCV '06* (2006).

[51] Kiwiel, K. Proximal minimization methods with generalized bregman functions. *SIAM Journal on Control Optimization 35* (1997).

[52] Klein, S., Staring, M., and Pluim, J. Evaluation of optimization methods for non-rigid medical image registration using mutual information and b-splines. *IEEE Transactions on Image Processing* (2007).

[53] Kohli, P., Shekhovtsov, A., Rother, C., Kolmogorov, V., and Torr, P. On partial optimality in multi-label mrfs. *Proc. of ICML '08* (2008).

[54] Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Analysis and Machine Intelligence 28* (2006).

[55] Kolmogorov, V., and Zabih, R. What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence 26* (2004).

[56] Komodakis, N. *Optimisation Algorithms for Discrete Markov Random Fields, with Applications to Computer Vision.* PhD thesis, Computer Science Department, University of Crete, 2006.

[57] Komodakis, N. Towards more efficient and effective lp-based algorithms for mrf optimization. *Proc. of ECCV '10* (2010).

[58] Komodakis, N., Paragios, N., and Tziritas, G. Mrf energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence 33* (2011).

[59] Komodakis, N., and Tziritas, G. Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Analysis and Machine Intelligence 29* (2007).

[60] Kumar, M., Kolmogorov, V., and Torr, P. An analysis of convex relaxations for map estimation of discrete mrfs. *Journal of Machine Learning Research 10* (2009).

[61] Kumar, M., Torr, P., and Zisserman, A. Solving markov random fields using second order cone programming relaxations. *Proc. of CVPR '06* (2006).

[62] Kwon, D., Lee, K., Yun, I., and Lee, S. Nonrigid image registration using dynamic higher order mrf model. *Proc. of ECCV '08* (2008).

[63] Larsson, T., Patriksson, M., and Stromberg, A. Ergodic primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming 86* (1999).

[64] Lewis, R. M., and Nash, S. G. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing 26*, 6 (2005), 1811–1837.

[65] Li, S. Z. *Markov Random Field Modelling in Image Analysis.* Advances in Computer Vision and Pattern Recognition. Springer-Verlag, 2009.

[66] Luong, D. V. N., Parpas, P., Rueckert, D., and Rustem, B. Solving mrf minimization by mirror descent. In *Advances in Visual Computing*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.

[67] Luong, D. V. N., Rueckert, D., and Rustem, B. Incorporating hard constraints into non-rigid registration via nonlinear programming. *Proc. SPIE. Medical Imaging: Image Processing* (2011).

[68] Malfait, M., and Roose, D. Wavelet-based image denoising using a markov random field a priori model. *IEEE Transactions on Image Processing 6* (1997).

[69] MODERSITZKI, J. *Numerical methods for image registration.* Numerical Mathematics and Scientific Computation. Oxford University Press, 2004.

[70] MODERSITZKI, J. Flirt with rigidity image registration with a local non-rigidity penalty. *International Journal on Computer Vision 76* (2008).

[71] MURAMATSU, M., AND SUZUKI, T. A new second-order cone programming relaxation for max-cut problems. *Journal of Operations Research of Japan 43* (2003).

[72] N. KOMODAKIS, N. P., AND TZIRITAS, G. Mrf optimization via dual decomposition: Message-passing revisited. *Proc. of ICCV '07* (2007).

[73] NASH, S. A multigrid approach to discretized optimization problems. *Optimization Methods and Software 14* (2000).

[74] NASH, S. G., AND LEWIS, R. M. Assessing the performance of an optimization-based multilevel method. *Optimization Methods and Software 26*, 4-5 (2011), 693–717.

[75] NEDIC, A., AND OZDAGLAR, A. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization 19* (2009).

[76] NEMIROVSKI, A., AND YUDIN, D. *Problem complexity and Method Efficiency in Optimization.* Wiley, 1983.

[77] NESTEROV, Y. *Introductory Lectures on Convex Optimization.* Kluwer, 2004.

[78] NESTEROV, Y. Smooth minimization of non-smooth functions. *Mathematical Programming 103* (2005).

[79] NESTEROV, Y. Gradient methods for minimizing composite objective function. *Mathematical Programming 140*, 1 (2013), 125–161.

[80] NESTEROV, Y., AND NEMIROVSKI, A. *Interior-Point Polynomial Algorithms in Convex Programming.* Studies in Applied and Numerical Mathematics. SIAM, 1994.

[81] NOCEDAL, J., AND WRIGHT, S. *Numerical Optimization.* Springer Series in Operations Research. Springer-Verlag, 2006.

[82] PAIGE, C. C., AND SAUNDERS, M. A. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM Journal on Numerical Analysis 12* (1975).

[83] Parpas, P., and Webster, M. A stochastic multiscale model for electricity generation capacity expansion. *European Journal of Operational Research 232*, 2 (2014), 359 – 374.

[84] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[85] Pennec, X., Cachier, P., and Ayache, N. Understanding the demons algorithm 3d non-rigid registration by gradient descent. *Medical Image Computing and Computer-Assisted Intervention* (1999).

[86] Ravikumar, P., Agarwal, A., and Wainwright, M. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research 11* (2006).

[87] Ravikumar, P., and Lafferty, J. Quadratic programming relaxations for metric labelling and markov random field map estimation. *Proc. of ICML '06* (2006).

[88] Rohlfing, T., Maurer, C. R., Bluemke, D. A., and Jacobs, M. A. Volume preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint. *IEEE Transaction on Medical Imaging 22* (2003).

[89] Rother, C., Kumar, S., Kolmogorov, V., and Blake, A. Digital tapestry. *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2005).

[90] Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., and Hawkes, D. Non-rigid registration using free-form deformations. *IEEE Transactions on Medical Imaging 18* (1999).

[91] Rustem, B. A class of superlinearly convergent projection algorithms with relaxed stepsizes. *Applied Mathematics and Optimization 12*, 1 (1984).

[92] Savchynskyy, B., Schmidt, S., Kappes, J., and Schnorr, C. A study of nesterovs scheme for lagrangian decomposition and map labeling. *Proc. of CVPR '11* (2011).

[93] Schmidt, M. Ugm: Matlab code for undirected graphical models. http://www.di.ens.fr/ mschmidt/Software/UGM.html, 2007.

[94] Schnabel, J., Rueckert, D., Quist, M., Blackall, J., Castellano-Smith, A., Hartkens, T., Penney, G., Hall, W., Liu, H., Truwit, C., Gerritsen, F.,

Hill, D., and Hawkes, D. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. *Proc. of MICCAI 2001 2208* (2001).

[95] Sherali, H., and Choi, G. Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs. *Operations Research Letters 19* (1996).

[96] Shor, N. *Minimization methods for nondifferentiable functions.* Springer, 1985.

[97] Sontag, D., Globerson, A., and Jaakkola, T. *Introduction to Dual Decomposition for Inference.* Optimization for Machine Learning. The MIT Press, 2011, ch. 1.

[98] Sra, S., Nowozin, S., and Wright, S. J. *Optimization for Machine Learning.* Neural Information Processing Series. The MIT Press, 2012.

[99] Staring, M., Klein, S., and Pluim, J. P. W. A rigidity penalty term for nonrigid registration. *Medical Physics* (2007).

[100] Su, M., and Xu, H.-K. Remarks on the gradient projection algorithm. *Journal of Nonlinear Analysis and Optimization* (2010).

[101] Sun, J., Shum, H. Y., and Zheng, N. N. Stereo matching using belief propagation. *Proc. of ECCV '02* (2002).

[102] Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. on Pattern Analysis and Machine Intelligence 30* (2008).

[103] Tappen, M., and Freeman, W. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *Proc. of ICCV '03* (2003).

[104] Taylor, C., and Bhusnurmath, A. Solving image registration problems using interior point methods. *Computer Vision ECCV* (2008).

[105] Thiagarajan, J., Ramamurthy, K. N., and Spanias, A. Learning stable multilevel dictionaries for sparse representation of images. *IEEE Trans. on Neural Networks and Learning Systems (Under review)* (2013).

[106] Thirion, J. Image matching as a diffusion process: An analogy with maxwells demons. *Medical Image Analysis* (1998).

[107] Veksler, O. Stereo correspondence by dynamic programming on a tree. *Proc. of CVPR '05* (2005).

[108] Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. Non-parametric diffeomorphic image registration with the demons algorithm. *Medical Image Computing and Computer-Assisted Intervention* (2007).

[109] Wainwright, M., Jaakkola, T., and Willsky, A. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing 14* (2004).

[110] Wainwright, M., Jaakkola, T., and Willsky, A. Map estimation via agreement on trees: Message-passing and linear programming. *IEEE Trans. on Information Theory 51* (2005).

[111] Weiss, Y., and Freeman, W. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transaction on Information Theory 47* (2001).

[112] Wen, Z., and Goldfarb, D. A line search multigrid method for large scale nonlinear optimization. *SIAM Journal on Optimization 20* (2009).

[113] Werner, T. A linear programming approach to max-sum problem: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence 29* (2007).

[114] Winkler, G. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*, vol. 27 of *Stochastic Modelling and Applied Probability.* Springer-Verlag, 2003.

[115] Yang, A., Zhou, Z., Balasubramanian, A., Sastry, S., and Ma, Y. Fast l1-minimization algorithms for robust face recognition. *IEEE Trans. on Image Processing 22* (2013).