A Discrete Graph Laplacian for Signal Processing

Panganai Gomo

EECE, School of Engineering

University of Birmingham

MPhil Electronic and Electrical Engineering

To mom, dad and the boys....

Acknowledgements

Firstly I would like to acknowledge my Supervisor, Dr. Mike Spann, for allowing me the freedom to explore my intellectual curiosities. His advice and direction is much appreciation.
Secondly my Father, Professor Zvenyika Alfred Gomo and my mother Mrs. Barbara Gomo require a special mention. Professor Gomo for being the first scientist I have known and for his constant encouragement in my intellectual pursuits and to Mrs. Gomo for her valuable lessons in efficacious and pragmatic approaches to life but mostly for their financial support.
Martin Schlueter for intellectual discussions in a coffee shop in Birmingham city and lastly to a string of people who deserve more mention than the listing of their names: Father Hugh Ross (Floss) for maths tutorship, Dawn Bannister for encouraging me to appreciate the visual arts and my brothers (Rodrigue and Jean-Christophe included) and my friends who have kept me sane (?) over the years.

A.M.D.G

Abstract

In this thesis we exploit diffusion processes on graphs to effect two fundamental problems of image processing: denoising and segmentation. We treat these two low-level vision problems on the pixel-wise level under a unified framework: a graph embedding. Using this framework opens us up to the possibilities of exploiting recently introduced algorithms from the semi-supervised machine learning literature.

We contribute two novel edge-preserving smoothing algorithms to the literature. Furthermore we apply these edge-preserving smoothing algorithms to some computational photography tasks. Many recent computational photography tasks require the decomposition of an image into a smooth base layer containing large scale intensity variations and a residual layer capturing fine details. Edge-preserving smoothing is the main computational mechanism in producing these multi-scale image representations. We, in effect, introduce a new approach to edge-preserving multi-scale image decompositions. Where as prior approaches such as the Bilateral filter and weighted-least squares methods require multiple parameters to tune the response of the filters our method only requires one. This parameter can be interpreted as a scale parameter. We demonstrate the utility of our approach by applying the method to computational photography tasks that utilise multi-scale image decompositions.

With minimal modification to these edge-preserving smoothing algorithms we show that we can extend them to produce interactive image segmentation. As a result the operations of segmentation and denoising are conducted under a unified framework. Moreover we discuss how our method is related to region based active contours. We benchmark our proposed interactive segmentation algorithms against those

based upon energy-minimisation, specifically graph-cut methods. We demonstrate that we achieve competitive performance.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In preparing this thesis I am reminded of the wisdom of the Ecclesiastic philosophers who spoke the words

> "*In searching many books there is no end*"

and of the words of William Blake

> "*Improvements make straight roads; but the crooked roads without improvements are roads of Genius.*"

spoken in his "*Proverbs from Hell*". Indeed the subject of image processing is a testament to these word; a vast subject touching many esoteric domains from mathematics, physics and computer science. Image processing can be considered as a branch of computer vision. Philosophically

> "computer vision *is a branch of science that tries to answer the philosophical, psychological and technical question: How can the local brightness information arriving at the retina (or to any optical sensor) be transformed into a global percept of the object, with their distance, color and shape?* "[Alvarez & Morel (1994a)].

In the context of computer vision, image processing is an input output process that receives the brightness information and transforms and extracts information that may be useful for higher level cognitive tasks. A viable approach to the image processing problem may be to model and mimic the behaviour of the

optical sensor of a biological organism, for example the human retina. A second approach may be to ignore the model of the image sensor and to work in a mathematical framework that produces the following outputs:

- A smooth (noise free) image from which reliable features may be extracted.
- A decomposition of the image domain into homogenous regions and boundaries.

These outputs are then passed on further down the computer vision chain where higher level cognitive tasks such as object recognition, depth perception or relative object motions can be recovered. The first output is what we consider an image denoising problem, which is very closely related to the subject of signal processing. The second output is considered an image segmentation problem.
These problems of image denoising and segmentation have substantial practical interest. Digital images, for example, may be degraded by noise such as sensor noise affecting images acquired in low lighting conditions. With digital cameras being ubiquitous in the consumer market it is not unusual for consumers to have encountered this problem. Image segmentation finds applications in video and image editing as well as medical image analysis. Even though these problems have been studied in the literature for many decades they still remain difficult and challenging. From a computer vision perspective these tasks are extremely challenging as they are mathematically ill-posed. We are required to estimate information or make inference about the likely clean image or desirable segmentation given the observed image. These problems are challenging as the space of possible likely outcomes is prohibitively large. It has long been recognised, in computer vision, that prior information is required to constrain the set of possible outcomes [Poggio *et al.* (1985)]. As such we are tasked with the problem of introducing models that capture some prior information about the image and allow us to achieve our desired task.

1.1 A mathematical setting

There is a diverse literature on models that seek to introduce prior information into the aforementioned low-level vision problems. To achieve very good per-

formance on a specific application many models of prior information have been specialised making it difficult to apply them to different tasks. Here we discuss a framework that uses diffusion processes, from mathematical physics, to achieve success in image processing tasks. We favour such a framework as it allows us to tackle both the denoising and segmentation tasks. The prior information that these models are built on can be stated as

pixels in a region are locally smooth,

that is brightness information is roughly constant in regions or image patches. This prior can be seen to be a relaxation of the more general statement:

pixels in local neighbourhoods are self-similar.

A prehistory of this setting is founded in the scale space theories and variational partial differential equation formulation of the 1980's and the early 1990's. In these theories an image can be expressed at multiple scales by the action of a blurring operator, such as

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla^2 u & t > 0 \\ u = f & t = 0. \end{cases} \tag{1.1}$$

where $t \geq 0$ is an abstract parameter known as the scale parameter, f is the original image at scale $t = 0$ and u is the image at scale $t > 0$. In mathematical physics ∇^2 is known as the Laplacian and defines a diffusion process. In image processing this is a blurring or smoothing process; hence this scale space smooths image data at ever greater values of the scale parameter t.

> "*Mathematically speaking, scale spaces are hierarchical decompositions/representaions at a continuum of scales embedding the image into a family of gradually more simplified versions* "[Bresson *et al.* (2006)].

A linear scale-space representation can be described by the following axioms [Perona & Malik (1990a)]

1. **Causality**. A scale-space representation should have the property that no spurious detail should be generated passing from finer to coarser scales.

2. **Immediate Localisation.** At each resolution, region boundaries should be sharp and coincide with the semantically meaningful boundaries at that resolution

3. **Piecewise smoothing.** At all scales intra-region smoothing should occur preferentially over inter-region smoothing.

The mathematical methods that generate scale spaces generally, but not exclusively, are differential equations and diffusion processes. Naturally a formal context for such descriptions is sought. It is in the series of papers [Sochen *et al.* (1995), Kimmel *et al.* (1997), Kimmel *et al.* (2000)] that a formal context for image processing scale-spaces was related to methods of high-energy physics, particularly String theories, using the mathematical language of differential calculus. In this area of high energy physics an elegant mathematical language is introduced to generalise notions of length, for the *world line*, and area, for the *world sheet*, through the action of a mathematical functional known as the *Polyakov action* measuring smoothness of high dimensional manifolds. Naturally these outputs that are desirable from an image processing point of view, specifically in image segmentation applications. This analogy to high-energy physics is still a young theory hence under constant development and scrutiny albeit it is a formal setting in which to develop image processing algorithms. Even though this theoretical setting is elegant it has pragmatic deficiencies. The image processing problem is inherently discrete due to the the nature of the acquisition process. As such the continuous partial differential equation operators have to be approximated by discrete analogues. This introduce a few practical problems including

- discretisation errors
- approximation of boundary conditions
- computational efficiency and stability problems.

As discrete operators are approximations of there continuous analogues their is inherently a discretisation error introduced in their formulation. Secondly in forming the discrete analogue we have to decide how the partial differential equations behave on the boundary of the image. Finally there is a trade off between

the computational efficiency and stability of the numerical solvers used to produce the solution. Albeit this setting is elegant and what we glean from it is the advantages introduced by considering geometric formulations of the underlying partial differential equations on the manifold. In this setting the manifold structure and geometric properties are important for deriving the image processing algorithms.

1.2 Graph Embedding and Manifold Learning

Very recently a variety of machine learning algorithms stemming from statistical and geometric theory have been proposed for dimensionality reduction of high dimensional data. These algorithms embed the discrete data points as the vertices of a graph. The algorithms are motivated by desiring to capture the geometric relationships inherent in the data. Of particular interest is the Laplacian Eigenmaps algorithm [Belkin & Niyogi (2001)] that captures self-similarities within the data. This algorithm draws on the correspondence of the graph Laplacian and the Laplacian operator on a continuous manifold. Justification of the algorithm comes from the notion that the Laplacian generated from the data points may be viewed as an approximation to the Laplace operator defined on the continuous manifold. This connection is well known to specialists in spectral graph theory [Chung (1997)] and we seek, in this thesis, to exploit this relationship in order to derive diffusion processes to effect image denoising and segmentation. Secondly we find that using this setting opens us up to the possibilities of exploiting algorithms from the semi-supervised machine learning literature. We will see that such algorithms are useful for adding constraints to the image segmentation problem through interactive user input. These algorithms may also be exploited in application areas such as computational photography. This is not the first time that graph embedding has been used in image processing on the pixel-wise level but the approach is significantly different to those proposed in [Shi & Malik (2000)].

1.3 Motivation

In this thesis we seek to exploit diffusion processes on graphs to effect image denoising, segmentation and enhancement. We seek to treat these low-level vision problems on the pixel-wise level under one framework. The algorithms are summarised using probabilistic graphical models as this allows us to

1. understand the processes in a probabilistic manner
2. summarise the algorithms in an energy minimisation framework which is currently de-facto in computer vision
3. discuss how to introduce higher-level prior knowledge of images in order to build more "intelligent" algorithms
4. avoid the practical problems of discretisation errors, introduction of boundary conditions and computational efficiency associated with the continuous analogues.

The framework we pursue is a graph-theoretic framework. As mentioned using this setting opens us up to the possibilities of exploiting machine learning algorithms from the semi-supervised learning and dimensionality reduction literatures. In this thesis we operate exclusively on the pixel-wise level as opposed to region patches or regions.
From the literature it is understood that image processing problems are inherently discrete. Energy-based minimisation methods using Markov random fields exploit this fact and as a result have grown in popularity and utility. However there is a gap in the literature with respect to graph theoretic methods. Recently [Shi & Malik (2000)], for example, introduced the normalised cuts algorithm for image segmentation using graph theoretic methods. They did not generalise to image processing problems in general. [Grady (2006)] built upon these ideas by recognising that the graph Laplacian used in the method of [Shi & Malik (2000)] are analogous to those of continuous partial differential equations. His motivations were to develop a discrete calculus with application to image processing and pattern recognition problems in general. In a series of papers [Grady (2006),Grady & Schwartz (2006),Couprie *et al.* (2009),Grady & Alvino (2009)]

he develops these ideas. It is upon these works we build upon. In particular we appreciate the Laplacian plays a central role in the theory of diffusion processes both for partial differential equations and graph theoretic methods. It is to this theory that we seek to make contributions. Moreover we seek, as much as possible, to make links to the Markov random field and probabilistic energy-based minimisation literature as such methods are de facto in computer vision.

1.3.1 Thesis Summary and Contributions

We summarise the motivations of the graph theoretic literature as applied to image processing in the statement:

> *Graph based diffusion processes provide a rich detail preserving scale-space for the development of image based pattern recognition and manipulation algorithms.*

This thesis details argument in favour of this statement. Moreover we could say image based pattern recognition algorithms could include image segmentation whilst manipulation algorithms could include denoising amongst other applications in computational photography. Support for this assertion is provided in this thesis as follows:

Chapter 2 reviews diffusion processes, graph-theoretic methods and prior modeling as applied to low-level image processing problems; particulary that of denoising. The chapter begins with a general introduction to prior modeling in low level vision using energy-based methods. A review of the literature on diffusion processes, graph theoretic methods, non-local models and graphical models is then presented. For completeness we discuss methods that seek to model images as a combination of basis elements. Finally we discuss the connections between these methods and the state-of-the-art.

Chapter 3 reviews diffusion process and energy-based methods. This chapter enables us to highlight the challenges associated with developing segmentation algorithms using continuous differential operators. The chapter concludes

with novel computational experiments that we published in the conference paper [Gomo & Spann (2009)] and concludes with a discussion of practical difficulties framed around

- discretisation errors
- approximation of boundary conditions
- computational efficiency and stability problems.

Chapter 4 studies diffusion processes on graphs as applied to the image denoising problem. Although the problem of image denoising is already well established and studied within the literature we tackle the problem from a new direction: through a graph theoretic approach. We review the recently proposed graph-based differential geometric arguments and recover the key innovations. Building upon these ideas we propose an alternate approach using graphical models and Markov random fields from which we derive a novel image denoising algorithm. We link the algorithm with the de facto energy minimisation methods in computer vision using the Markov random field. Moreover we interpret the new algorithm using nonparametric statistics and kernel density estimation.
The proposed algorithm is mathematically equivalent to the Google PageRank algorithm [Brin & Page (1998),Page *et al.* (1999)] for ranking hypertextual world wide web documents where by the associated Markov process is generated from the photometric similarities between image pixels.
Specifically this chapter contributes the innovations:

- a novel image denoising algorithm using graph-theoretic arguments from which preliminary results were published in [Gomo (2010)]
- a formulation of the algorithm using Markov random fields
- a novel derivation of the Google PageRank algorithm for ranking hypertextual world wide web documents.

Chapter 5 carries out a comprehensive evaluation of the PageRank denoising algorithm. We propose that the PageRank denoising algorithm is a solution of

the Poisson equation on a graph. To facilitate the evaluation we develop another novel edge-preserving smoothing algorithm. This algorithm, which we call Power Iteration denoising (PID), is based upon the iterative application of the normalised pair-wise similarity matrix generated from the image data. Further evaluations of these algorithms are carried out by applying them to computational photography tasks that require edge-preserving smoothing algorithms. We compare and benchmark with de facto and state-of-the-art methods.
Specifically this chapter contributes the innovations:

- a novel edge-preserving smoothing algorithm which we call Power iteration denoising - experiments and theory pertaining to this algorithm were accepted as a conference paper at the 9^{th} *International Conference on Machine Learning and Applications (ICMLA'10)* entitle "*Power Iteration Denoising*"
- analytic and experimental arguments that show that the PageRank denoising algorithm in the solution of the Poisson equation on a graph.
- analytic and experimental arguments that show that the Power Iteration denoising algorithm in the solution of the Laplace equation on a graph.
- thorough experimental evaluations and applications to computational photography are provided. These experiments and applications were summarised in a *EUROGRAPHICS* paper entitled "*Edge-Preserving Image Decompositions using the Random Walk Graph Laplacian*" currently under review.

Chapter 6 shows that with very minor modifications to the graph-based diffusion algorithms: Power Iteration denoising and PageRank denoising, we can derive interactive image segmentation algorithms. These algorithms are related to methods from semi-supervised machine learning and active contour methods.
Specifically this chapter contributes the innovations:

- two novel interactive image segmentation algorrithms
- experimental evaluations and benchmarking with de facto state-of-the-art methods.

Chapter 7 summarises our contributions and support for our main thesis claim. We conclude the thesis with a discussion on further applications and future work. Specifically evidence and support for the central claim of this thesis is provided by the key contributions

1. the development of two novel edge-preserving filter algorithms
2. application of the edge-preserving filter algorithms to problems in computational photography
3. extension of the associated diffusion processes to interactive (semi-supervised) image segmentation

which are elaborated upon and further discuss in the final chapter.

Throughout this thesis we use an electrical network analogy to give intuitive arguments to the algorithms that we develop. This analogy uses arguments based upon Ohm's and Kirchoff's current laws. This is not arbitrary as [Perona & Malik (1990b)] provide similar arguments to provide intuition to their scale-space axioms and anisotropic image diffusion algorithms. Further to this semi-supervised machine learning algorithms are often described with such analogies [Zhu *et al.* (2003a)] providing a good link between these methods.

Chapter 2

Models for Image Processing

2.1 Chapter Summary and Contributions

In this chapter we seek to introduce and review diffusion processes, graph-theoretic methods and prior modeling as applied to low-level image processing problems; particulary that of denoising. We begin the chapter with a general introduction to prior modeling in low level vision using energy-based methods. A review of the literature on diffusion processes, graph theoretic methods, non-local models and graphical models is then presented. For completeness we discuss methods that seek to model images as a combination of basis elements. Finally we discuss the connections between these methods and the state-of-the-art.

2.2 Introduction

From the point of view of a cognitive science, image processing may be considered a tool to extract information from an image. This information may prove useful for recovering relative shapes and order, recovering geometry and topologies, studying motions and patterns or even the removal of unwanted fine grain detail that may be considered noise. From an engineering point of view this low level process may be used to design and construct higher level cognitive machines able to recognise objects, determine relative motions in scenes or filtering of unwanted information from images. Many energy-minimisation methods.

Much of the progress made recently in image processing may be attributed to the development of models of image content and their pragmatic employment in their relevant applications. Developed models span from simple l_2 norm smoothness terms, to edge preserving norms such as total variation filters, to models that seek to capture the self-similarities in the image to sparse and redundant image representations. The research, development and evolution of models is at the very heart of the image processing literature. One may ask what is a model and why do we need one? This may be answered by considering the problem of image denoising

> *Given a noisy image acquired from an imaging sensor; one is required to obtain the 'clean' denoised image.*

To provide a separation of the noisy image content from the 'true' image we are clearly required to have some familiarity with the process that generated it. This is called the *forward model*. Computer graphics problems are usually associated with simulating this forward model. In image processing a typical formulation of this forward problem is to assume the noisy image is generated by an additive noise process; that is

$$f = u + \epsilon \tag{2.1}$$

where f is the observed image, u is the original image and ϵ is an additive noise process with finite mean, μ, and variance, σ^2. The inverse problem is to recover the clean image u from the noisy observation f. A classic way to recover the noisy signal is to attempt to minimise the l_2 distance

$$|f - u|^2, \tag{2.2}$$

known as the *data-fidelity* term, subject to some constraints that reflect our prior information or belief of the nature of the original image, that is a *regularisation* term. A classical example is to assume that near-by pixels in 'well-behaved' images exhibit strong correlations; that is they are spatially smooth in localised regions. This is a model that can be expressed mathematically as

$$|\nabla u|_2^2 \tag{2.3}$$

and is a constraint. ∇ is interpreted as a derivative operator. Therefore to recover the image one can formulate the problem as seeking the value of u that minimises

$$\frac{1}{2}\int |\nabla u|_2^2 d\Omega + \frac{\lambda}{2}|f-u|_2^2 d\Omega \tag{2.4}$$

where the first term is a regularisation term, the second term is the data fidelity term, Ω is the image domain and λ is a regularization parameter to be specified. The regularisation parameter balances the trade off between the regularisation term and the data fidelity term. This is the variational calculus form of the denoising problem. This problem may also be formulated in a Bayesian inference sense

$$p(u|f) \propto p(f|u)p(u) \tag{2.5}$$

where we seek to minimize the log-likelihood function., $p(f|u)$ is the likelihood function (data-fidelity) term and $p(u)$ is the prior probability or model of prior information (regularization term) of u. This principle is generally known as *maximum a posterior* estimation. We have neglected the evidence term $p(f)$ which ensures the posterior probability

$$p(u|f) = \frac{p(f|u)p(u)}{p(f)} \tag{2.6}$$

integrates to unity. To perform the optimisation we may seek a function that minimises the negative log-likelihood function

$$-\log p(u|f) \propto -\log p(f|u) - \log p(u). \tag{2.7}$$

In general Variational calculus has been used to model many problems of mathematical physics. The variational calculus can trace it's routes to the Brachistrone minimal path problems studied by Johann Bernoulli and his brother Jakob circa 1696. The method was later formalized by the Swiss mathematician and physicist Leonhard Euler and his student Joseph-Louis de Lagrange in the mid 1700's. Since this time the variational calculus has become an important tool for modeling in physics, science and engineering being applied to continuous optimization problems in heat flow, diffusion processes in molecular dynamics, optics, elasticity theory, fluid dynamics and semiconductor modeling amongst a myriad of

other areas. A review of the calculus of variations applied in image processing can be found in [Chan *et al.* (2003) and references there in]. Typically calculus of variations has been applied to image processing problems such as motion estimation and optical flow [Horn & Schunck (1982)], gradient vector flow [Xu & Prince (1998)], image inpainting (image interpolation) [Esedoglu & Shen (2002) and references there in], image denoising and debluring [Rudin *et al.* (1992),Chambolle (2004a),Aujol (2009) and references there in] and image segmentation [Osher & Sethian (1988),Mumford & Shah (1989),Chan & Vese (2001),Osher & Fedkiw (2003)]. The variational and Bayesian inference methods are equivalent in philosophical motivation and can be summarised as energy based functionals

$$E(\mathbf{u}, \mathbf{f}) = E_r(\mathbf{u}) + E_d(\mathbf{u}, \mathbf{f}) \tag{2.8}$$

where the first term $E_r(\mathbf{u})$ is the regularisation term that models our prior belief of the behaviour of the data and $E_d(\mathbf{u}, \mathbf{f})$ is the data fidelity term that fits our model to the observed data.

2.2.1 From Continuous Models to Discrete Codes

The calculus of variations approach to modeling image processing problems begins with a continuous model of the image. An optimization problem is then setup. Continuous optimization is carried out by the techniques of the variational calculus. An optimality solution can be designated by the *Euler-Lagrange* equations. These continuous partial (sometimes ordinary) differential equations can then be cast into a discrete form using finite difference methods. The resulting finite difference system is solved by a linear or nonlinear solver as appropriate. This continuous model of image processing tends to model the images as functions in a regular function space such as the Hilbert space $\mathbb{H}(\Omega)$ or the space of functions of bounded variation, $\mathbb{BV}(\Omega)$, where discretisation of the problem is only considered in the context of numerical implementation on a digital computer. Inherently the nature of a digital image is that of a discrete set of pixels. Recently many authors have thought it natural to model the problem discretely.

> *To make it clear the problem is stated as a discrete model not simply for numerical implementation.*

Naturally a discrete framework to model the problem is that of an undirected graph [Chung (1997)]. Indeed recently this has become an active thread of research in image processing circles. The seminal work [Shi & Malik (2000)] formulated the problem of image segmentation as that of spectral graph partitioning [Pothen *et al.* (1990)] solving an eigenvalue problem to recover the graph partitioning (image segmentation). In this setting the vertices of a graph correspond to image pixels and to pairs of pixels (edges) a weight is associated dependent on a similarity measure. Continuing with this line of thought other authors proposed improvements and extenstions to this framework for image segmentation of note [Fowlkes *et al.* (2004), Grady & Schwartz (2006)] whilst other methods used diffusion processes and lazy random walks on graphs [Grady (2006)].
Of late if has been recognised that this framework is in general a potentially powerful model for solving other image processing problems seemingly unrelated to graph partitioning. For example [Zhang & Hancock (2006)] proposed an image denoising model based upon the diffusion process on a graph. This method is known to be related to heat diffusion on a *Riemannian manifold*. It would appear almost simultaneously and independently that other researchers realised the power of formulating image processing problems on a graph, for instance deblurring and denoising [Kindermann *et al.* (2005)], image segmentation [Sumengen & Manjunath (2006), Grady & Alvino (2009)] and general inverse problems [Peyré *et al.* (2008)]. These models have a strong connection to data representation models in machine learning [Belkin & Niyogi (2001),Zhou & Schlkopf (2004),Nadler *et al.* (2005),Hein *et al.* (2005),Coifman & Lafon (2006)].
From a different perspective image processing algorithms were being developed in the discrete setting under the theory of Markov random fields (MRFs) [Bishop (2007),Roth & Black (2009)]. These methods, although related to graph models, are formulated from a different perspective. Instead of using the calculus of variations to derive optimisation strategies they use Bayesian inference. One of the biggest challenges in using MRF formulations is the development of efficient algorithms that find the required low-energy solutions. Over the last few years a variety of algorithms were developed for finding these low-energy solutions. Representative methods include simulated annealing [Besag (1986)], graph cuts

[Greig *et al.* (1989),Boykov & Jolly (2001),Komolgorv & Zabih (2004)] and belief propagation [Yedidia *et al.* (2003),Felzenszwalb & Huttenlocher (2004)]. The choice of the inference technique is known to greatly influence the performance of these models in applications. In computer vision applications the graph cuts method is possibly the de facto method of choice.

2.3 Continuous Models for Image Processing

In this setting we can consider the image a function in 3 dimensional space. We formulate the problem on the unit square Ω bounded by $[0,1] \times [0,1] \mapsto \mathbb{R}^2$ with the coordinates (x_1, x_2). The 3^{rd} dimension is given by the mapping of the domain of Ω given by the function $u = u(x_1, x_2)$. Many image processing problems can

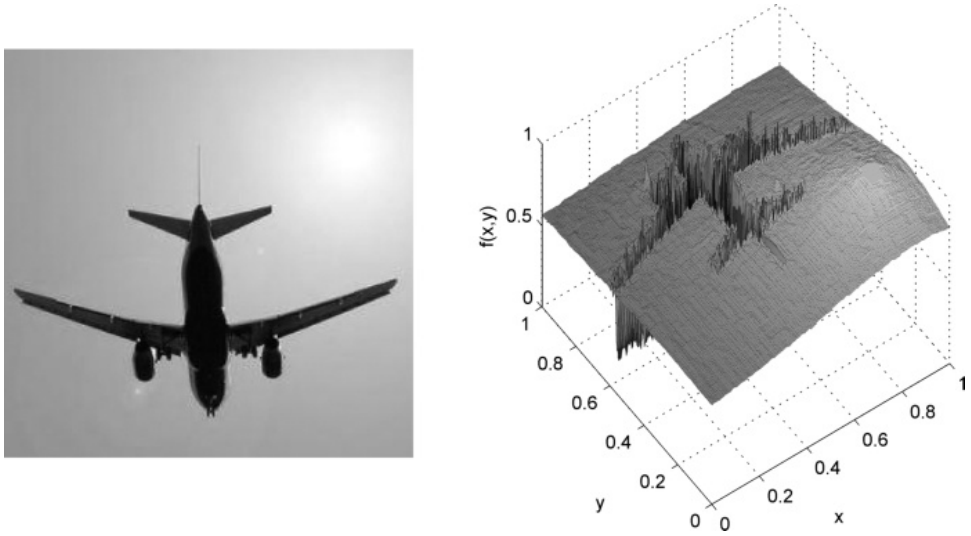


Figure 2.1: Illustration of the surface representation of an image. The image on the left can be considered as a two dimensional embedded in 3 dimensional space as represented by the right image.

be modeled by the action of a linear function on the image subject to a noise process

$$f = K * u + \eta \tag{2.9}$$

where K is a linear operator accounting for blurring, $*$ is the convolution operator and η is some additive noise. The observed image is f whilst the original image is u. The goal of image recovery is to recover the original image from the observed data. To recover the required information the Euclidean distance between the observed image and the output is minimised

$$\frac{1}{2}\int_{\Omega} |K * u - f|^2 d\Omega. \tag{2.10}$$

This will not recover the original image as this problem is ill-posed. In general this is an *inverse problem* and a *regularisation* term is required to impose some constraints on the image. Early applications of inverse problems in image processing was to solve the denoising problem where the linear constraint

$$J(u) = \int_{\Omega} |\nabla u|^2 d\Omega \tag{2.11}$$

is imposed to minimize the distance

$$\frac{1}{2}\int_{\Omega} |f - u|^2 d\Omega \leq \sigma^2 \tag{2.12}$$

where σ^2 is the standard deviation of the noise in the model. This model assumes smoothness of u, for instance, the derivatives of u or the local variation $|\nabla u|$ is small. This scheme is a linear *Tikhonov* regularization scheme. One could use a model where the regulariser assumes u to be a function of bounded variation, that is the regularisation term is finite

$$J(u) = \int_{\Omega} |\nabla u| d\Omega < \infty. \tag{2.13}$$

Minimisation of the resulting optimisation problem

$$\int_{\Omega} |\nabla u|^2 d\Omega + \frac{1}{2}\int_{\Omega} |f - u|^2 \tag{2.14}$$

can be carried out using the calculus of variations leading to the diffusion process

$$\nabla^2 u + f - u = 0. \tag{2.15}$$

This elliptic partial differential equation is the Euler-Lagrange equation that minimises equation (2.14). As with partial differential equation models, boundary

conditions must be specified. The Neumann boundary condition, for example, are usually sufficient for image processing problems. They state that the amount of image intensity in the domain is conserved and can be stated as

$$\frac{\partial u}{\partial \vec{n}} = 0. \tag{2.16}$$

that is

$$\frac{\partial}{\partial t}\left[\int_{\Omega} u(x_1, x_2, t) d\Omega\right] = 0. \tag{2.17}$$

This diffusion model can be traced back to an idea proposed by Denis Gabor [Lindenbaum & Fischer (1994)] who proposed the difference between the original image and observed image is roughly proportional to the Laplacian

$$\frac{K_h * u - f}{h} \propto \nabla^2 u \tag{2.18}$$

where K_h is the blurring kernel with scale parameter h. In the denoising problem the blurring Kernel is the identity dirac function. As $h \to 0$ the model resembles a linear isotropic diffusion equation

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla^2 u & t > 0 \\ u = f & t = 0. \end{cases} \tag{2.19}$$

These ideas form the basis of the scale-space theory of vision [Witkin (1983), Perona & Malik (1990b),Alvarez & Morel (1994a), Weickert *et al.* (1999)] which propose a multiscale representation of images. This multiscale representation is achieved by applying the simple diffusion model of equation (2.19). In general analytic closed form solutions of partial differential equations do not exist but for this equation the analytic solution is given by [Lindenbaum & Fischer (1994), Koenderink (1984), Weickert *et al.* (1999)]

$$\begin{cases} u = G_{\sqrt{2t}} * f & t > 0 \\ u = f & t = 0. \end{cases} \tag{2.20}$$

where G_σ is a Gaussian kernel with standard deviation σ explicitly written as

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{(-\frac{x_1^2 + x_2^2)}{2\sigma^2}}. \tag{2.21}$$

The higher the standard deviation the greater the blurring. As the $\sigma \to 0$ the Gaussian kernel tends to a dirac delta function. The Gaussian convolution model produces an equivalent solution to the diffusion process of equation (2.19) where $\sigma = \sqrt{2t}$ [Lindenbaum & Fischer (1994)].

This formulation using inverse problems, calculus of variations and partial differential equations has a direct analogy to methods from mathematical physics. In particular the diffusion model of equation (2.19) plays a big role. This is a diffusion process on a flat Euclidean space. This analogy runs deep as one can consider the image intensity as a physical variable, such as temperature and considering the amount of heat in a domain is conserved[1] we have the general diffusion equation

$$\frac{\partial u}{\partial t} =< \nabla \cdot (D \nabla u) > \tag{2.22}$$

where D is a diffusion tensor characterising the diffusion of heat within the medium. The diffusion tensor can be used to introduce an anisotropic flow to the heat diffusion model of equation (2.19). This is the idea of the Perona-Malik type partial differential equation filters [Perona & Malik (1990b)]. The image denoising filters introduced in their paper seek to remove high frequency noise (Gaussian) whilst preserving edges. For their diffusion matrix they choose the scalar

$$D = g(|\nabla f|^2) \tag{2.23}$$

where g is a monotonically decreasing function such that

$$g(z) = \frac{1}{1 + \beta z} \tag{2.24}$$

and $g(0) = 1$, $g(z) \to 0$ as $z \to \infty$. The image diffusion law is given by

$$\frac{\partial u}{\partial t} =< \nabla \cdot g(|\nabla f|^2) \nabla u > . \tag{2.25}$$

Practical solutions to this problem choose the diffusivity function g as the edge indicator function

$$g(|\nabla f|^2) = \frac{1}{1 + \beta |\nabla (G_\sigma * f)|^2}. \tag{2.26}$$

[1] The Neumann boundary condition.

[Weickert (1998)] decided to take into consideration the direction of localised structure for the diffusion model instead of the edge indicator function. The diffusion tensor D is then decomposed into its eigenvectors and eigenvalues

$$D = \phi \Lambda \phi^T \tag{2.27}$$

such that $\phi = (\mathbf{v}_1, \mathbf{v}_2)$ is the matrix of the normalised eigenvectors as it's columns ad Λ is the diagonal matrix with the eigenvalues of D along it's principal diagonal and D is taken to be

$$D = \begin{pmatrix} 1 + f_x^2 & f_x f_y \\ f_x f_y & 1 + f_y^2 \end{pmatrix}. \tag{2.28}$$

A total variation minimisation formulation can be motivated as follows: Returning to the inverse problem stated in equation (2.14) the optimization strategy can be considered as a minisation for u:

$$\lambda J(u) + \frac{1}{2} \int_\Omega (f - K * u)^2 d\Omega \tag{2.29}$$

where $J(u)$ is chosen as the total variation norm $J(u) = \int_\Omega |\nabla u| d\Omega$, K is the identity or delta function and λ is a parameter that controls the influence of the regularization parameter and is dependent on the noise. The variational calculus is utilised to obtain optimality conditions. This model denoises the image whilst preserving edges. The associated Euler-Lagrange equation is given by

$$\frac{\partial u}{\partial t} =< \nabla \cdot (\frac{\nabla u}{|\nabla u|}) > +\lambda(f - u). \tag{2.30}$$

This model can be thought of as a weighted diffusion process where the local weighting is inversely proportional to the local variation of the image hence in regions of low variation (smooth) $\frac{1}{|\nabla u|}$ is large and diffusion is not impeded. In regions where the local variation is large, such as near edges, $\frac{1}{|\nabla u|}$ is small and diffusion is impeded. One can notice that this total variation filter is similar to the Perona-Malik type filters with

$$g = \frac{1}{|\nabla u|}. \tag{2.31}$$

More recently the partial differential equation filters have been formalised in a framework borrowed from *high-energy physics*, particularly *string theory*, in the series of papers [Sochen *et al.* (1995), Kimmel *et al.* (1997), Kimmel *et al.* (2000)] [1]. The authors show links to the classical scale spaces and image diffusion processes through this model. In such a setting grey level images are considered as 2 dimensional surfaces embedded in a 3 dimensional space. *Differential geometry* is an essential tool in the construction of this model and the derivation of algorithms for denoising images in this setting. In differential geometry the notion of distance on a non-flat manifold is key. If we consider a grey level image as being indexed by local surface coordinates [2] (σ^1, σ^2) and the image intensity is given by the feature coordinate $f(\sigma^1, \sigma^2)$ then the triple describe a manifold in 3 dimensional Euclidean space

$$(\sigma^1, \sigma^2, f(\sigma^1, \sigma^2)). \tag{2.32}$$

Now the notion of distance in 3 dimensional Euclidean space is familiar so the arclength can be written as

$$ds^2 = (d\sigma^1)^2 + (d\sigma^2)^2 + df^2 \tag{2.33}$$

and the mapping

$$\mathbf{u} : \Sigma \rightarrow M \tag{2.34}$$

can be introduced which maps the 2 dimensional surface Σ, indexed by local coordinates (σ^1, σ^2) onto the 3 dimensional manifold M by the coordinate transform

$$\mathbf{u} : (u^1(x_1, x_2), u^2(x_1, x_2), u^3(x_1, x_2)). \tag{2.35}$$

Then the induced distance metric on the 2 dimensional image surface is the pullback from the manifold M having the bilinear form

$$ds^2 = g_{\mu\nu} d\sigma^\mu d\sigma^\nu \tag{2.36}$$

[1] String theory seeks to unify the four basic forces of physics and reconcile gravitational theory and quantum mechanics.

[2] We have not assumed Cartesian although this coordinate system is allowable.

and using the Einstein summation convention [Nakahara (1990)]

$$ds^2 = g_{11}(d\sigma^1)^2 + g_{12}d\sigma^1 d\sigma^2 + g22(d\sigma^1)^2. \tag{2.37}$$

By the pairs (Σ, g) we denote the 2 dimensional image surface and its metric tensor and by (M, h) the manifold and its metric tensor [1] respectively. Since an embedding space has been defined an action functional on the space of embedding functions is to be defined. The action on the mapping can be expressed as

$$S[\mathbf{u}, \Sigma, M] = \int d^{m_\Sigma}\sigma\sqrt{g}g^{\mu\nu}\partial_\mu u^i \partial_\nu u^j h_{ij}(\mathbf{u}) \tag{2.38}$$

again using the Einstein summation convention. This is the most general form for measuring area on non-Euclidean manifolds where m_Σ denotes the dimension of Σ, g is the determinant of the image metric, $g^{\mu\nu}$ is the inverse of the image metric $\partial_\mu u^i = \frac{\partial u^i}{\partial \sigma^\mu}$ and h_{ij} is the metric tenor on M. This action for $m = 2$ was first proposed in string theory eponymously named the *Polyakov action* [Polyakov (1981)]. Given this action functional one has to choose the variable for which it is minimised. Using the calculus of variation the Euler-Lagrange equation that minimise this functional is given by

$$-\frac{1}{2\sqrt{g}}h^{il}\frac{\delta S}{\delta X^l} = \frac{1}{\sqrt{g}}\partial_\mu(\sqrt{g}g^{\mu\nu}\partial_\nu u^i) + \Gamma^i_{jk}\partial_\mu u^j \partial_\nu u^k g^{\mu\nu} = 0. \tag{2.39}$$

where Γ^i_{jk} is the Levi-Civita connection [Nakahara (1990)]. The interested reader can find the derivation in [Sochen *et al.* (1995)].
To gain an intuitive feel of the action functional S one may choose, for example, to vary only the feature coordinate in Euclidean space and fix the metric $g_{\mu\nu}$ and other coordinate such that

$$g = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; h_{ij} = \delta_{ij}, \; x_1 = \sigma^1, \; x_2 = \sigma^2 \tag{2.40}$$

and varying the feature coordinate

$$u(x_1, x_2) \tag{2.41}$$

[1] Recall that in Euclidean spaces the metric tensor is the delta function

the we get, up to a non important constant, the action functional

$$S[\mathbf{u}, \Sigma, M] = \int d^2\sigma |\nabla u|^2 \tag{2.42}$$

which can be recognised as the standard l_2 Tikhonov regulariser for continuous functions. Minimising this functional produces the standard Laplacian operator. The choices of parameters to vary and metric tensors produces different Laplacians which are important for image denoising problems, that is the linear scale space, the scale space of Perona-Malik and the total variation flow amongst others can be recovered. Again fixing $x_1 = \sigma^1$ and $x_2 = \sigma^2$ and leaving $g_{\mu\nu}$ free, with the embedding space M being Euclidean, then the minimisation of the Polyakov action for grey level images yields the Beltrami flow

$$u_t = \Delta_g u \tag{2.43}$$

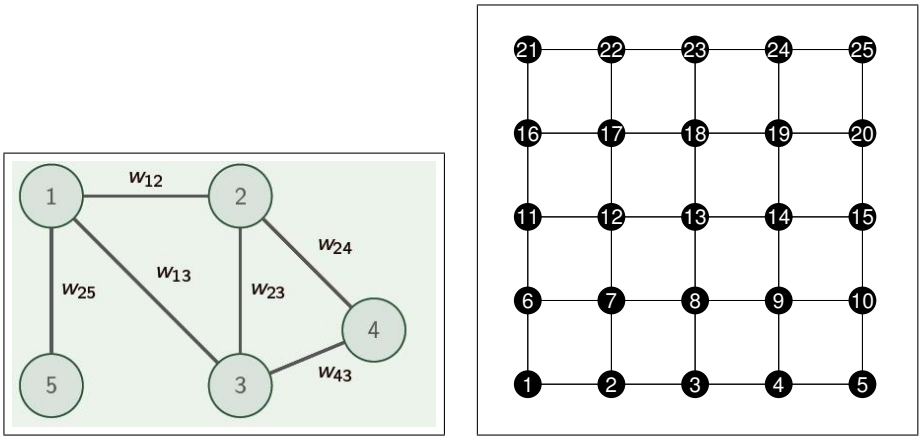
where Δ_g is known as the Laplace-Beltrami operator which is a generalisation of the Laplacian operator from flat Euclidean spaces to non-flat manifolds and is explicitly given by

$$\Delta_g = \frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu u). \tag{2.44}$$

2.4 Discrete Models for Image Processing

Graph based methods for image processing have attracted a considerable amount of attention over the last decade since the publication of the normalised cuts algorithm for image segmentation [Shi & Malik (2000)]. Rather than focus on local image features and their consistency these approaches aim to extract global impressions of the image though pairwise similarities of image pixels. These models share a philosophy with a class of algorithms for ranking hypertextual world wide web documents. The ranking algorithms seek to rank the authority of a web page document not by the content of the individual web page but extract a global view by constructing models of the links between web pages. The most successful of these algorithms (commercially) drives the Google web search engine [1] [Brin & Page (1998), Page *et al.* (1999)].

[1] www.google.com



(a) (b)

Figure 2.2: Illustration of the graph model for Image processing. (a) Illustrates a general graph. (b) Illustrates the natural horizontal ordering for a 5×5 image patch.

In the discrete setting, to represent the image data each pixel element is labeled as the vertex (node) of an undirected graph $G = \{V, E\}$ with a set of nodes V and a set of edges E. An edge e_{ij} exists between a set of pixels if a certain connectivity requirement is satisfied. Figure 2.2 (a) illustrates the general notion of a graph whilst figure 2.2 (b) illustrates a $4-$connected lattice structure for a 5×5 pixel image patch. The values of the pixels elements are then stored in the vector

$$\mathbf{f} = \begin{pmatrix} f(x_{25}) \\ f(x_{24}) \\ \vdots \\ f(x_2) \\ f(x_1) \end{pmatrix}. \tag{2.45}$$

To each edge of the graph a weighting function, $w_{ij} = w(f(x_i), f(x_j))$, is associated. The measure is usually positive preserving and symmetric

$$w_{ij} \geq 0, \;\; w_{ij} = w_{ji}. \tag{2.46}$$

This weighting is determined by a similarity measure with the pixels in a neighbourhood region, for example a $4-$connected or $8-$connected lattice structure. Edge weights control the flow or diffusion of information across the graph. If the values are large then heat can flow through the nodes easily. If the edge weighting is low then flow across the node is impeded. This has an intimate connection with random walk models and transition probabilities. The degree of a vertex is the sum of the weights of it's associated edges

$$d_i = \sum_j w_{ij}. \tag{2.47}$$

The degree matrix D is defined as the matrix with the edge weights along the main diagonal and the weight matrix

$$W_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ w_{ij} & \text{if } i = j \end{cases}$$

Now we can define the *graph Laplacian*, this is given by

$$L_{ij} = \begin{cases} d_i - w_{ij} & \text{if } i = j \\ -w_{ij} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.48}$$

Unfortunately there is no one graph Laplacian [1]. The normalised cuts algorithm constructs the following Laplacian

$$L_n = D^{-1/12} L D^{-1/12}. \tag{2.49}$$

The spectral decomposition of the normalized Laplacian $L_n = \Phi\Lambda\Phi^T$, $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ then reveals important information of the structure of the image, for example, in image segmentation problems the second smallest eigenvalue known as the *Fiedler* and its associated eigenvector are sought. They provide information that can be used to partition and segment the image. In particular the eigenvalue problem

$$L_n \mathbf{v} = \lambda \mathbf{v} \tag{2.50}$$

[1] In chapter 4.4.3 we discuss various graph Laplacians.

is solved. [Meila & J (2001)] reinterpreted the eigenvalue problem by choosing the Laplacian

$$L_{rw}\mathbf{v} = \lambda\mathbf{v} \tag{2.51}$$

where they proved the eigenvalues are consistent with the original normalised cuts model. It was not until the publication [Zhang & Hancock (2006)] that graph based models were applied to the image denoising problem. They proposed to denoise the image by the graph Heat equation

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -L_n\mathbf{u} & t > 0 \\ \mathbf{u} = \mathbf{f} & t = 0. \end{cases} \tag{2.52}$$

L_n is the graph Laplacian and we can note the analogy to the continuous model of equation (2.52). They proposed a solution based upon a spectral decomposition and the heat kernel and discard the iteration

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -L\mathbf{u}^n \tag{2.53}$$

where the initial data is the observed image $\mathbf{f}$. We can note the analogy with Gabor's idea for diffusion filtering and scale space (equations 2.18 and 2.19). In their paper they could not justify a suitable choice for the scale/time parameter Δt and a stopping time.

2.5 Non-local Models

The Bilateral filter [Tomasi & Manduchi (1998)] is a non iterative method that updates a pixel location by a weighted average of pixels in a neighbourhood. The weighting is based upon the geometric proximity and photometric similarity between pixels favouring near by pixels as opposed to distant ones. The continuous representation of the bilateral filter can be stated as

$$h(x_1, x_2) = \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\hat{\mathbf{x}})c(\hat{\mathbf{x}}, \mathbf{x})s(f(\hat{\mathbf{x}}), f(\mathbf{x}))d\hat{\mathbf{x}}}{K} \tag{2.54}$$

where K is a normalisation

$$K = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(\hat{\mathbf{x}}, \mathbf{x})s(f(\hat{\mathbf{x}}), f(\mathbf{x}))d\hat{\mathbf{x}} \tag{2.55}$$

where $\mathbf{x} = (x_1, x_2)$ and $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$ are cartesian coordinates and $f(\cdot)$ is the image intensity. This term contains the product of two functions c and s which represent the geometric proximity and photometric similarity respectively. In implementations these functions are Gaussian kernels

$$c(\hat{\mathbf{x}}, \mathbf{x}) = e^{(\frac{-d(\hat{\mathbf{X}}, \mathbf{X})}{2\sigma_d})^2}, \; s(\hat{\mathbf{x}}, \mathbf{x}) = e^{(\frac{-d(f(\hat{\mathbf{X}}), f(\mathbf{X}))}{2\sigma_r})^2} \tag{2.56}$$

where $d(\hat{\mathbf{x}}, \mathbf{x})$ is the Euclidean distance, for example

$$d(\hat{\mathbf{x}}, \mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|. \tag{2.57}$$

The parameter σ_d is the blur radius. Greater values of σ_d result in greater blurring of the image and if too large may fail to preserve high frequency details. The parameter σ_r determines contrast blurring, lower parameters correspond to preservation of contrast. Implementation is by an approximation of the integral operator as a discrete sum. The integral is taken over a finite window. The scheme is non iterative.

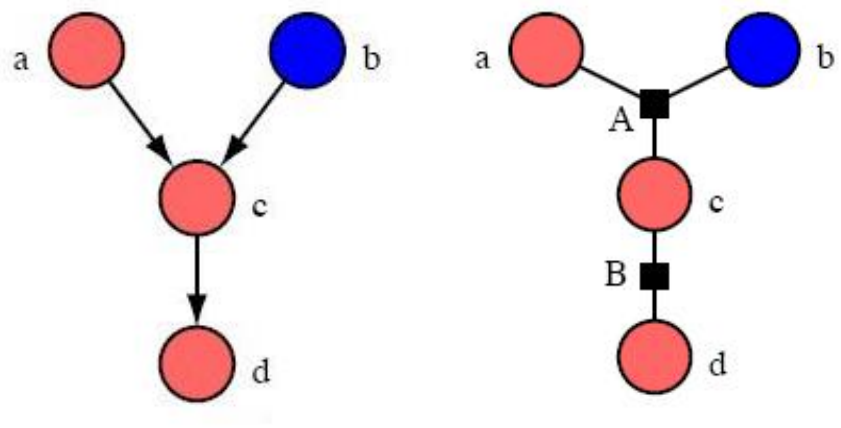
The non-local means filter [Buades *et al.* (2005a)] draws from these ideas and uses the self-similarities in the images as a prior on the natural images. It hypothesises that the information required to model an image pixel is encoded in the self-similarities in the pixels defined in a neighbourhood. If we consider a noisy image, $\mathbf{f}$, then the recovered denoised pixel elements can be obtained as

$$u_i^{n+1} \leftarrow \sum_{u_{ji}} \frac{K_\sigma(u_i^n - u_j^n)}{\sum_{u_j \sim N(u_i)}(u_i^n - u_j^n)} u_j^n \tag{2.58}$$

where K_σ is a Gaussian kernel and $\mathbf{u}^0 = \mathbf{f}$. We notice that the recovered image intensity is independent of the current value and uses only the information from similarities with it neighbours. This is the simplest case of the non-local means filter and even though rediscovered in the literature it can be attributed to [Lee (1983)] and Yaroslasky's neighbourhood filters[Buades *et al.* (2005a)].

This formulation is just an application of the Nadaraya-Watson kernel estimator from the pattern recognition literature [Fukunaga & Hostetler (1975),Cheng (1995),Bishop (2007)]. In this measure of similarity the kernels

$$K_\sigma = e^{\frac{-|u_i^n - u_j^n|^2}{\sigma^2}} \tag{2.59}$$



(a) (b)

Figure 2.3: Directed graphical models. Dark blue node indicates an observed random variable whilst the red indicated hidden random variables. Figure (a) is a directed graphical model. Figure (b) is an undirected graphical model.

are close to 0 for dissimilar pixels and close to 1 for similar pixels. In effect the similarity measures are used to generate a weighted average of pixels within a neighbourhood.

2.6 Graphical Models

In order to understand further the probabilistic implications of the graph-based methods we introduce graphical models. A thorough review is beyond the scope of this thesis so here we only introduce the salient aspects that allows us to describe our graph-based methods with respect to graphical models. A thorough discussion can be found in [Bishop (2007)]. Graphical models allow us to understand how the nodes of the graph interrelate in a probabilistic manner through geometric intuition. These models have been successfully applied in diverse areas including natural language processing, computational biology, robot navigation and computer vision. A graphical model $G = \{\mathbb{V}, \mathbb{E}\}$ associates with each node of the graph $f(x_i)$ a random variable of the vector $\mathbf{f}$. Nodes of a graph can either be observed or hidden, in which case they need to be inferred. Figure 2.3 (a) shows a directed graphical model. In a directed graphical model all the edges are

directed. These models are also known as Bayesian networks. If we defined the set of parent nodes $\pi(i)$ indicating the set of all nodes having an edge directed into f_i we can write the probability distribution under this model as

$$p(\mathbf{f}) = \Pi_{f_i \in \mathbb{V}} p(f_i | \mathbf{f}_{\pi(i)}) \tag{2.60}$$

where $\mathbf{f}_{\pi(i)}$ are the components of the random vector belonging to the parents of f_i. So the probability distribution for figure 2.3 (a) can be factorised as

$$p(a, b, c|b) = p(d|c)p(c|a, b)p(a). \tag{2.61}$$

Even though this model is known to have nice properties it is not used in this thesis. It is the undirected graphical model that is of interest and used to summarise the probabilistic nature of our graph constructions. In a directed graphical model all the edges are directed and these models are known as Bayesian networks. Undirected graphical models do not pose such restrictions. With these models we associate a potential function with each clique $\Psi_k : \mathbf{f}_{(k)} \rightarrow \mathbb{R}$ assigning a positive score to the random variables of a clique. A clique is a set of nodes of a graph where each is a neighbour of the others. The joint probability of the distribution can be expressed as

$$p(\mathbf{f}) = \frac{1}{Z} \Pi_{k=1}^{K} \Psi_k(\mathbf{f}_k) \tag{2.62}$$

for a graph with K cliques. Z is a normalisation term ensuring the distribution integrates to unity. This model is often expressed equivalently as a Gibbs distribution

$$p(\mathbf{f}) = \frac{1}{Z} \exp \left[- \sum_{k=1}^{K} U_k(\mathbf{f}_{(k)}) \right]. \tag{2.63}$$

The equivalence is justified by the Hammersly-Clifford theorem [Roth & Black (2009)]. Abusing terminology U_k is also called a potential function. We call the function

$$E(\mathbf{f}) = \sum_{k=1}^{K} U_k(\mathbf{f}_{(k)}) \tag{2.64}$$

the energy of the distribution. Now the normalisation term is defined as

$$Z = \int_{\mathbf{f}} \Pi_{k=1}^{K} \Psi(\mathbf{f}_{(k)}) d\mathbf{f} = \int -\sum_{k=1}^{K} U_k(\mathbf{f}_k) d\mathbf{f}. \tag{2.65}$$

Computation of this normalisation term can be intractable as one is required to sum over the space of all random images $\mathbf{f}$. This aspect makes learning and inference in undirected graphical models difficult. Figure 2.3 (b) shows an example of an undirected graphical model. The black squares indicate cliques to which we can assign potential functions U_A and U_B.
Undirected graphical models are termed Markov random fields when the two following requirements are fulfilled

- positivity $p(\mathbf{f}) > 0$
- Markov property $p(f_i|\mathbf{f}_{V\backslash\{i\}}) = p(f_i|\mathbf{f}_{n(i)})$

where $V\backslash\{i\}$ is the set of all nodes except for that indexed by i and $n(i)$ is the set of all neighbours of i.
Inference using these models can be carried out in a Bayesian framework where we seek the MAP estimate of

$$p(\mathbf{u}|\mathbf{f}) \propto p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \tag{2.66}$$

where the graphical model is used to impose prior information into the model. In the denoising literature, state-of-art performance in the application of graphical models and Markov random fields to the problem was due to the introduction of what is known as the fields-of-experts model [Roth & Black (2009)] where the image prior

$$p(\mathbf{u}) = \Pi_{k=1}^{K} \Psi(\mathbf{f}_{(k)}) \tag{2.67}$$

is chosen so that the potential is the product of expert distributions [Welling *et al.* (2002)]

$$\Psi(\mathbf{f}_{(k)}) = \Pi_{m=1}^{M} \phi(\mathbf{J}_m^T \mathbf{f}_k; \alpha_m) \tag{2.68}$$

where J_m is considered a linear filter defined over a clique, ϕ is an expert probability distribution to be specified and α_m are the parameters of the expert distribution. The number of expert distributions is given by M. The full image prior can be expressed as

$$p(\mathbf{u}) = \frac{1}{Z}\left[\Pi_{k=1}^{K}\Pi_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{u}_{(k)};\alpha_m)\right] \tag{2.69}$$

and by the Clifford-Hammersly theorem can equivalently be expressed in the Gibbs form as

$$p(\mathbf{u}) = \frac{1}{Z}\exp\left[\sum_{k=1}^{K}\sum_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{u}_{(k)};\alpha_m)\right]. \tag{2.70}$$

MAP estimation is carried out by finding the vector $\mathbf{u}$ that minimises the energy

$$E(\mathbf{u}|\mathbf{f}) = -\frac{1}{2}(\mathbf{f}-\mathbf{u})^T\Sigma^{-1}(\mathbf{f}-\mathbf{u}) + \sum_{k=1}^{K}\sum_{m=1}^{M}(\mathbf{J}_m^T\mathbf{u}_{(k)};\alpha_m). \tag{2.71}$$

2.7 Sparse and Redundant Image Representations

On a different track of research it has been noted that images (or image patches) can be represented as a combination of elements in a basis. This basis is known as a dictionary. For exmaple

$$\mathbf{f} = \alpha_1\mathbf{d}_1 + \alpha_2\mathbf{d}_2 + \ldots\alpha_m\mathbf{d}_m \tag{2.72}$$

where d_i are the elements of a dictionary, basis elements or atomic images and the coefficients α_i are to be determined. The dictionary elements may be known a-priori; such as a discrete cosine transform basis (DCT), a discrete wavelet transform (DWT) basis or a basis learned from an image database. Here we consider the vector $\mathbf{f}$ as an $\sqrt{n}\times\sqrt{n}$ image patch lexicographically ordered as an $N\times 1$ vector where $N = \sqrt{n}\times\sqrt{n}$. We can express this basis decomposition as a linear system

$$\mathbf{f} = D\mathbf{a} \tag{2.73}$$

where the columns of D contain the dictionary elements $D = [\mathbf{d}_1, \mathbf{d}_2, \ldots \mathbf{d}_m]$ and is known as the the dicitonary matrix and the vector $\mathbf{a}$ contains the coefficients such that $\mathbf{a} = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T$.
In the sparse and redundant models the vector $\mathbf{a}$ is sparse; that is $\mathbf{a}$ has k non-zero elements where $k << m$. The sparsity of the vector $\mathbf{a}$ can be measured by the l_0 norm $\|\mathbf{a}\|_0^0$ which is defined as the number of non-zero elements of the vector $\mathbf{a}$. The set of possible combinations of elements $k << m$ chosen to represent the image $\mathbf{f}$ is given by

$$C_m^k = \frac{m!}{k!(m-k)!}. \tag{2.74}$$

Initially this image model was used in an atomic decomposition framework [Chen *et al.* (2001)]. The goal was to solve the problem

$$\min \|\mathbf{a}\|_0^0 \text{ s.t. } \|\mathbf{d} - D\mathbf{a}\|_2^2 \leq \delta \tag{2.75}$$

which is a combinatorial optimisation problem that seeks the sparsest vector $\mathbf{a}$, with coefficients $\alpha_i = 1$, that explains the image $\mathbf{f}$ in terms of elements of the dictionary D. This problem gives rise to greedy algorithms such as the Method of Frames (MOF), Matching Pursuit and Best orthogonal Basis (BoB) [Chen *et al.* (2001)] to name a few. The difficulty in solving this problem is that these strategies perform a combinatorial optimisation over prohibitively large search spaces. More pragmatic solutions to this problem arise from relaxing the regularising term from $\|\mathbf{a}\|_0^0$ to $\|\mathbf{a}\|_1^1$ giving the problem

$$\min \|\mathbf{a}\|_1^1 \text{ s.t. } \|\mathbf{f} - D\mathbf{a}\|_2 \leq \delta. \tag{2.76}$$

A detailed description and review of optimisation strategies can be found in [Yang *et al.* (2010)]. Returning to the denoising problem at hand; for an appropriate choice of the dictionary we may seek to solve the problem

$$\min \lambda \|\mathbf{a}\|_p^p + \frac{1}{2}\|\mathbf{f} - D\mathbf{a}\|_2^2 \tag{2.77}$$

where the choice of $p = 0$ and $p = 1$ lead to the optimisation procedures of equations (2.76) and (2.77) respectively.
The dictionary D can be chosen to be a wavelet basis (DWT) leading to an

iterative shrinkage algorithm. The discrete wavelet transform of an image $\mathbf{f}$ is represented by the multiplication

$$\mathbf{a} = T\mathbf{f} \tag{2.78}$$

where T is an orthogonal matrix and the appropriate wavelet dictionary to be used is chosen as

$$D = T^T \tag{2.79}$$

and therefore

$$\|D\mathbf{a} - \mathbf{f}\|_2^2 = \|T^T\mathbf{a}\mathbf{f}\|_2^2 = \|\mathbf{a} - T\mathbf{f}\|_2^2 \tag{2.80}$$

and therefore the denoising process translates to finding the vector $\mathbf{a}$ that minimises

$$\lambda\|\mathbf{a}\|_p^p + \frac{1}{2}\|\mathbf{a} - \mathbf{b}\|_2^2 \tag{2.81}$$

where $\mathbf{b} = T\mathbf{f}$ is the discrete wavelet transform of the noisy image and the final image is recovered as

$$\mathbf{u} = T^T\mathbf{a}. \tag{2.82}$$

The optimisation decouples into a set of independent scalar problems with closed form solutions for the cases $p = 0$ and $p = 1$ known as hard- and soft- thresholding respectively. In this setting the solution of the denoising problem is obtained by a component wise soft/hard threshold applied to image data in the wavelet domain. The recovered image is obtained by transforming the vector $\mathbf{a}$ of coefficients back to the image domain.

This image processing strategy is known as a *transform-shrink-inverse tansform* approach. A more recent applcation of sparse and redundant representation methods to image denoising [Elad & Aharon (2006)] requires one to select D as a dictionary learned from an image database. Instead of working on the entire image the sparse and redundant model is applied on image patches for example $\sqrt{n} = 8$ [Elad & Aharon (2006)]. Every patch in the given image is expected to have a sparse representation with respect to the dictionary D. Thus for an 8×8 image

patch we have the dictionary matrix with dimensions 64×256.
The denoising is carried out by seeking the vectors $\mathbf{u}$ and $\mathbf{a}$ that minimise the program

$$\frac{1}{2}\|\mathbf{f}-\mathbf{u}\|_2^2+\sum_{i\in\Omega}\|\mathbf{a}_i\|_0^0 \text{ s.t. } \|R_i\mathbf{u}-D\mathbf{a}_i\|_2^2\leq\delta\forall\ i\in\Omega \tag{2.83}$$

where Ω is the image domain, $i\in\Omega$ is a patch index. The matrix operation R_i extracts an $\sqrt{n}\times\sqrt{n}$ patch from a location i and the sparse coefficients vector $\mathbf{a}_i$ is chosen to represent patch $R_i\mathbf{u}$ to within a specified tolerance. Unfortunately one has to find both the set of representations and the denoised image $\mathbf{u}$. Practically this is carried out by first fixing $\mathbf{u}=\mathbf{f}$ and finding the dictionary D and the representations $\mathbf{a}_i, i\in\Omega$. Once found $\mathbf{u}$ is recovered by solving

$$\min\frac{1}{2}\|\mathbf{u}-\mathbf{f}\|_2^2+\lambda\sum_{i\in\Omega}\|R_i\mathbf{u}-D\mathbf{a}_i\|_2^2. \tag{2.84}$$

2.8 Discussion

As we have seen in this chapter a variety of denoising methods have been proposed in the literature. Different methods for introducing prior information about the nature of the image to be recovered have been proposed. These models essentially share the same goal:

> *smooth noisy variations in the image whilst preserving meaningful structure.*

Here we would like to relate and review the important contributions of these models to the literature.

Diffusion Filtering, Gaussian Filtering and Graph Methods. Diffusion filtering and Gaussian filtering, methods are motivated by the assumption that images have parts that are locally smooth. The diffusion filtering methods exploit this assumption by diffusing (averaging) image intensities in localised regions according to a diffusion equation such as the linear isotropic scale-space law:

$$\begin{cases} \frac{\partial u}{\partial t}=\nabla^2 u & t>0 \\ u=f & t=0. \end{cases} \tag{2.85}$$

which is the basis upon which the scale-space theory of vision is built [Weickert *et al.* (1999)]. Although a closed form solution of this equation does not exist analytic solutions are given by the convolution of the original image with the Gaussian kernel at multiple scales. That is Gaussian filtering at multiple scales. It is well known in the literature that Gaussian convolution is equivalent to the diffusion on a manifold. These linear isotropic diffusion methods provide satisfactory results in regions that are locally smooth but tend to blur edges and textures. A large class of methods were introduced to overcome these problems including the anisotropic diffusion techniques [Perona & Malik (1990b)] and the total variation methods [Rudin *et al.* (1992), Aujol (2009)]. These diffusion filtering methods introduce some geometric notions, such as curvature based diffusion, into the diffusion process. Ever more complex formulations were introduced to describe these methods using the calculus of variations leading to the the high-energy physics formulations [Sochen *et al.* (1995)]. In general diffusion methods can be related to diffusion equations on graphs. It is well know to specialists in spectral graph theory that the graph Laplacian can be considered as an analogue to the Laplacian operator on the continuous manifold [Chung (1997)]. The advantage of utilising the graph Laplacian is that it captures geometric notions of the manifold. Although some results in the area of mesh and image processing have been recently published [Elmoataz *et al.* (2008), Bougleux *et al.* (2009)] this connection has not been fully exploited in low-level vision.

Graph methods, Graphical models Non-local methods. Although the application of graph methods in image processing have not been motivated by probabilistic notions [Shi & Malik (2000),Elmoataz *et al.* (2008)] we know a probabilistic connection can be sought by considering image processing problems as that of energy minimization using Markov random fields. Markov random field models have been applied to image denoising problems since the early 1980s [Geman & Geman (1984), Besag (1986)]. These models had substantial use but often produced results with strongly smoothed edges [Tsuzurugi J (2002)]. It was recently shown that the model of [Roth & Black (2009)] produced state-of-the-art results using Markov random field models. Although this model produces state-of-the-art results it needs to be trained over a large database of images. Further to this inference with this model is computationally expensive.

Non-local methods were not introduced as probabilistic models. They were introduced as methods that exploit the prior information that images are locally self-similar. The bilateral filters [Tomasi & Manduchi (1998)] and non-local means filter [Buades *et al.* (2005a)] that have found great utility due to the fact that they produce high quality results [Chatterjee & Milanfar (2010)], do not need to be trained and are not too expensive computationally. Albeit these models can be given a probabilistic interpretation; which we shall develop in this thesis - particularly for the non-local means method. If we consider the non-local means method we notice that this model is essentially a Markov random field. Recalling that Markov random fields have the requirements of positivity; which is implied by the kernel being positive preserving and the Markov property

$$p(u_i^n|\mathbf{u}_{V\setminus\{i\}}^{n-1}) = p(u_i^n|\mathbf{u}_{\mathcal{N}(i)}^{n-1}). \tag{2.86}$$

Realising our choice of kernel function has restricted us to pairwise cliques we have an underlying Graph structure $G = \{\mathbb{V}, \mathbb{E}\}$ such that $\mathbf{u}$ and $\mathbf{f}$ are indexed by the vertices of G and this model describes a *conditional random field* as the updated variable u_i^n is conditioned upon the self-similarities of the previous iteration $\mathbf{u}_{\mathcal{N}(k)}^{n-1}$ and obey the Markov property. This joint distribution can be expressed as the conditional random field [Lafferty (2001),Kumar & Martial (2003)].

$$p(u_i^n|\mathbf{u}_{\mathcal{N}(i)}^{n-1}) \tag{2.87}$$

Transformation and Wavelet Domain methods. This category of denoising algorithms has produced some of the most accurate methods. This category of methods work with the *transform-shrinkage-inverse transform* approach to image modeling which operates as follows: (1) decompose the image patches as a linear combination of basis elements; (2) modify the coefficients using a "shrinkage" or "coring" strategy and (3) reconstruct the denoised image. [Donoho (1995)] developed a shrinkage algorithm where wavelet coefficients are typically shrunk towards zero using a fixed threshold dependant on the noise levels in the image. Other wavelet strategies seek to model the marginal statistics of wavelet coefficients and use a Bayesian threshold [Simoncell (1999)] to effect denoising. More recent methods try to model the spatial dependencies of neighbouring coefficients [Portilla & Simoncell (2000), Portilla *et al.* (2003)]. Other sparse representation

methods use curvelets [Stark *et al.* (2002)] or even basis elements learned from the patches of natural images [Elad & Aharon (2006)]. It is the fusion of non-local means methods and transform-domain shrinkage methods that have yielded the best results across all categories of image denoising [Dabov *et al.* (2007),Chatterjee & Milanfar (2010)]. Lastly can be noted that an interesting relationship between denoising in the wavelet domain and diffusion filtering has been proposed [Steidl *et al.* (2004)].

Chapter 3

Image Segmentation

3.1 Chapter Summary and Contributions

In this chapter we seek to provide a brief review of a somewhat exhaustive subject. We discuss the problem of image segmentation. The emphasis is placed on methods inspired by diffusion processes and partial differential equations such as the active contour methods. The literature of which in itself is vast. We also provide a brief discussion on other important image segmentation methods. In the second half of this chapter we discuss the problems associated with using diffusion processes based segmentation algorithms; highlighting the problems alluded to in the introductory chapter of this thesis namely:

- discretisation errors
- approximation of boundary conditions
- computational efficiency and stability problems.

This discussion is augmented with experimental investigations into the numerical simulation of the gradient vector flow processes; a method for providing external forcing functions for active contour models.

3.2 Introduction

An important problem in image analysis and computer vision is the segmentation and modeling of shapes from images. Mathematical models and algorithms are required to automatically segment regions of information. More specifically to present a mathematical representation of semantic objects acquired from imaging devices. This is one of the desirable outputs of an image processing system as discussed in the introduction to this thesis. As a low-level vision problem we could say that image segmentation is the problem of

> *finding groups of pixels that "go together".*

In statistics such a problem would fall under the area of cluster analysis [Jain *et al.* (2004)]. Segmentation is one of the oldest problems in computer vision. The early techniques focussed on using region splitting and merging strategies [Brice & Fennema (1970),Horowitz & Pavlidis (1976)] which correspond to agglomerative algorithms in the cluster analysis literature [Jain *et al.* (2004)]. More recent algorithms tend to optimise a global criterion such as intra-region consistency or statistics and inter-region boundary lengths or dissimilarities. Such methods include the Mumford-Shah functional [Mumford & Shah (1989)], normalised cuts [Shi & Malik (2000)], mean-shifting [Comaniciu & Meer (2002)] and level sets with region statistics [Cremers *et al.* (2007)]. In this chapter we will review the active contour methods and briefly discuss graph based methods such as normalised cuts and graph cuts.

3.3 Active Contours

Within the last two decades since the seminal work of Kass-Witkin-Terzopoulos [Kass *et al.* (1988)] active contour models have become a very successful method for image segmentation and shape modeling. The method is a variational approach based upon the minimisation of a parametrised curve in space. The objective of the active contour model is to minimise the functional

$$P(C) = \alpha\frac{1}{2}\int_0^1 |C'(s)|^2 ds + \beta\int_0^1 |C''(s)|^2 ds + \int_0^1 F_{ext}(s)ds. \tag{3.1}$$

This model describes a parameterised dynamic curve in 2D or a surface in 3D. The parameters α and β are free parameters and $C(s) = [x_1(s), x_2(s)]$ for $s \in [0, 1]$. The first two terms of the model describe the internal forces of the curve whilst an external force $F_{ext}(s)$ attracts the curve to an image boundary. The Euler-Lagrange equations that minimise the functional (3.1) lead to the flow

$$C_t(s) = \alpha \frac{\partial^2 C}{\partial s^2} - \beta \frac{\partial^4 C}{\partial s^4} + \nabla_s f(C) = 0 \tag{3.2}$$

with prescribed boundary conditions. There are several inherent difficulties with this model. Amongst problems of choice of the free parameters and an accurate method for estimating the fourth order derivatives in the model we have to define an external forcing function. Nevertheless this model has found tremendous utility in a host of applications. Extensive research efforts have been focussed in reformulations of the active contour model (3.1) and defining the external forcing function.

A popular reformulation of the problem is that of the geodesic active contour [Caselles *et al.* (1997), Kichenassamy *et al.* (1995)]. The objective function associated with this model is given as

$$\int_0^{L(C)} g(|f(C(s))|) ds \tag{3.3}$$

where f is the original image and $L(C)$ is the Euclidean length of the image. Using the variational level set framework [Osher & Sethian (1988), Osher & Fedkiw (2003)] show that the flow resulting from the Euler-Lagrange equations can be written as

$$u_t = |\nabla u| g(\nabla f) \kappa + < \nabla g(\nabla f) \cdot \nabla u > \tag{3.4}$$

where

$$\kappa = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \tag{3.5}$$

is the Euclidean curvature and $g(\cdot)$ is a monotonically decreasing function defined on the positive half segment. Although this model has advantages over the original active contour model such as allowing curves to merge and split it still suffers

from similar deficiencies related to the definition of the external forcing function. In designing an external forcing function there are several desirable attributes. The external force should

1. improve the active contour model's robustness to initial conditions
2. aid the active contour model to describe complex boundaries
3. aid the active contour model to converge to a global minimum, avoiding local minima

3.3.1 Forcing Function

The original active contour models and geodesic active contour models were drawn to the edges by a potential function. This potential function is derived from the negative gradient direction of the image gradients [Kass *et al.* (1988), Caselles *et al.* (1997), Kichenassamy *et al.* (1995)]. Images can be thought of as non-smooth functions hence finding a global minimum is difficult. This results in the active contour models becoming trapped in local minima. Multiresolution techniques [Leroy *et al.* (1996)], heuristics such as genetic algorithms [Ballerini (2001)] and stochastic optimisation [Juan *et al.* (2003), Juan *et al.* (2006)] techniques have been employed to overcome these difficulties. These models do not directly address the image force but help to overcome local minima. To overcome sensitivity to initialisation and increase capture range balloon forces were introduced [Cohen (1991), Cohen & Cohen (1993)]. The breakthrough in defining image forces was in the introduction of the gradient vector flow (GVF) model [Xu & Prince (1998)]. A tremendous research effort has been focused on the design of the image forces. Not only is this model robust to local minima it also provides a wide capture range and aids convergence into concave boundaries. Further works in developing the concept of gradient vector flow and in general image vector flows have lead to models such as normalised gradient vector diffusion (NGVD) [Yu & Bajaj (2002)], curvature vector flow (CVF) [Gil & Radeva (2003)], flux maximising geometric flows [Alexander & Kaleem (2002)] and vector field convolution [Li & Acton (2007)] amongst others.
Of the methods in the literature, for defining an external forcing function, the

gradient vector flow method is possibly the most successful. The gradient vector flow is computed as a diffusion of the gradient vectors of a grey-level image or binary image derived from the image. The resulting gradient vector flow field provides a large capture range and forces the active contour into the concave regions. This force is more robust to initial conditions, is bidirectional and more robust to local minima. Although it provides many advantages we can still establish two deficiencies in this model

1. it has a considerably high computational cost
2. it smooths vectors across boundaries and does not respect geometry or localised structure

Gradient vector flow is computed as the steady state solution of a reaction-diffusion system. This is an initial value problem. The resulting flow is linear and isotropic resembling the linear Helmholtz equation. As a result it smooths across boundaries and does not respect localised structure. Solution of this system comes at a high-computational cost and does not scale well with image size. If numerical and storage efficient schemes are not used to generate the gradient vector flow then it may become difficult to simulate on large images. In the image processing community multigrid methods are commonly used to deal with the solution of large algebraic systems resulting from a variational partial differential equation. Indeed multigrid methods have a long established history in image processing dating back to works such as [Szeliski & Terzopoulos (1989)] to more recent works [Henn (2001), Han *et al.* (2007),Papandreou & Maragos (2007), Köstler *et al.* (2008),Strümer *et al.* (2008),Haber & Modersitzki (2006)]. A multigrid algorithm was even proposed for the numerical solution of the gradient vector flow problem [Han *et al.* (2007)]. Conjugate gradient methods have enjoyed some success in image processing problems of this type [Heers *et al.* (2001), Kohlberger *et al.* (2005)].

3.3.2 Region Statistics and Interactive Inputs

Active contour models are often susceptible to getting trapped in local minima even with the introduction of advanced forcing functions such as the gradient

vector flow. To overcome such problems one can augment the system with further higher level knowledge to constrain the set of possible solutions and reduce the effect of local minima. An interesting and effective strategy is to introduce region statistics into the energy minimisation framework of the active contour formulation. The statistical models may include statistics about colour, texture and motion interior and exterior to the region of interest. A recent review of methods using region statistics can be found in [Cremers *et al.* (2007)]. These methods are built upon the energy based ideas of the Mumford-Shah functional [Mumford & Shah (1989)] and the Chan-Vese flow [Chan & Vese (2001)].
Alternatively an oracle or domain expert may be used to interactively provide control points to guide the motion of the evolving contour or optimisation strategy. The intelligent scissors system developed in [Mortensen & Barrett (1995)] does just that. The interactive strategy is as follows

- the user supplies a rough outline or guide of the region to be segmented
- the system computes and draws a better segmentation.

Such a strategy is the philosophy behind current interactive segmentation algorithms [Boykov & Jolly (2001),Blake *et al.* (2004),Rother *et al.* (2004)].

3.4 Spectral Clustering and Normalised Cuts

Normalised cuts as introduced in [Shi & Malik (2000)] seek to group pixels by establishing clusters of similar pixels and separating clusters with weak affinities. Normalised cuts have emerged as an efficient method for grouping similar data and are easily applied to any data as long as a suitable similarity measure can be defined. The data is represented as a graph structure and at the heart of the algorithm is the search for the second smallest eigenvector associated with the graph Laplacian, hence it is a spectral clustering method.
Given a set of image pixels ordered lexicographically $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$ we can associate the pixels with a graph $G = \{\mathbb{V}, \mathbb{E}\}$. The objective of the partitioning algorithm is to split the data into disjoint sets

$$C_1, C_1, \ldots, C_k. \tag{3.6}$$

Initially we consider a bipartitioning problem where we define the cut as

$$cut(C, \bar{C}) = \sum_{i \in C, j \in \bar{C}} w_{ij} \tag{3.7}$$

where w_{ij} is understood to be the usual notion of weighting between nodes or pixels measuring the degree of similarity between pixels. This cut criteria is therefore a measure of the degree of dissimilarity between the clusters C and $\bar{C}$. An optimal bipartition is therefore a partition that minimises this value. As has been noted the minimum cut criteria favours cutting small sets of isolated nodes [Wu & Leahy (1993),Shi & Malik (2000)]. Therefore [Shi & Malik (2000)] proposed a more suitable measure known as the normalised cut criteria defined as

$$Ncut = \frac{cut(C, \bar{C})}{assoc(C, \mathbb{V})} + \frac{cut(C, \bar{C})}{assoc(\bar{C}, \mathbb{V})} \tag{3.8}$$

where $assoc(C, \mathbb{V}) = \sum_{i \in C, j \in \mathbb{V}} w_{ij}$ is the association (sum of weights) within the cluster and

$$\begin{aligned} cut(C, \bar{C}) \quad &= assoc(C, \mathbb{V}) - assoc(C, C) \\ &= assoc(\bar{C}, \mathbb{V}) - assoc(\bar{C}, \bar{C}). \end{aligned} \tag{3.9}$$

The normalised cut measure is a better partition criteria than the cut criteria as it looks for collections of edges that are weak relative to all edges both inside and emanating from a particular region. Unfortunately such problems are known to be NP-complete. As such the normalisd cut problem is embedded into the real value domain where an approximate solution can be found efficiently. [Shi & Malik (2000)] propose to find the vector $\mathbf{u}$ that minimises the Rayleigh quotient

$$\frac{\mathbf{u}^T (D - W)\mathbf{u}}{\mathbf{u}^T D \mathbf{u}}. \tag{3.10}$$

This is known to be the solution of the generalised eigenvalue problem

$$(D - W)\mathbf{u} = \lambda D \mathbf{u} \tag{3.11}$$

and in general the eigenvalue problem

$$L\mathbf{u} = \lambda \mathbf{u} \tag{3.12}$$

where $L = D^{-1/2}(D - W)D^{-1/2}$. The solution that is sort is the vector $\mathbf{u}$ which is the second smallest eigenvector, also known as the Fiedler vector. The values associated with the positive and negative components of the vector $\mathbf{u}$ are then associated with the two partitions. Putting it all together the normalised cuts algorithm can be expressed in algorithmic form as in algorithm (1). The original

Algorithm 1 Normalised cuts (Shi and Malik)

1: Construct weight matrix W and degree matrix D
2: $L = D^{-1/2}(D - W)D^{-1/2}$
3: Solve $L\mathbf{u} = \lambda\mathbf{u}$ for the Fiedler vector
4: Decide if bipartition should be further subdivided
5: Recursively partition if so required
6: Use K-means algorithm to generate clusters $C_1, C_2, \ldots, C_k$

normalised cuts algorithm used spatial points and image features to compute pixel-wise similarities, for example the per-pixel weights could be generated as

$$w_{ij} = \exp\left[\frac{-\|\mathbf{F}_i - \mathbf{F}_j\|_2^2}{\sigma_F^2} - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma_s^2}\right] \tag{3.13}$$

for pixels within a spatial radius $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 < \tau$ and $\mathbf{F}$ is a feature vector consisting of intensities, or colours, or oriented filter histograms or mixtures.
Normalised cuts have been extended in many directions. [Meila & J (2001)] use the random walk Laplacian

$$L_{rw} = D^{-1}(D - W) \tag{3.14}$$

in stead of the normalised Laplacian in the eigenvalue problem of equation 3.12 relating their solution to random walks on graphs. [Fowlkes *et al.* (2004)] and [Sharon *et al.* (2006)] propose computationally efficient strategies to solve the associated eigenvalue problem at the heart of the normalised cuts problem. [Fowlkes *et al.* (2004)] use the Nyström extension whilst [Sharon *et al.* (2006)] use an approach inspired by algebraic multigrids [Briggs *et al.* (2000)]. [Zhang & Hancock (2006)] used this algorithm to effect edge-preserving smoothing.

3.5 Graph Cuts

Graph cut methods seek to group pixels that have similar appearance or statistics. The active contour methods were optimised using iterative gradient descent techniques, which are prone to getting trapped in local minima. To address such problems [Boykov & Jolly (2001)] proposed to apply a binary Markov random Field (MRF) optimisation algorithm initially developed by [Greig *et al.* (1989)] to effect a binary segmentation. The approach uses a min-cut/max-flow algorithm from combinatorial optimisation to achieve the binary segmentation. The problem can be expressed as that of searching for the class label vector $\mathbf{l}$ that minimises the energy

$$E(\mathbf{l}) = \sum_{i \in \mathbb{V}} D_i(l_i) + \sum_{(i,j) \in n} V_{i,j}(l_i, l_j) \tag{3.15}$$

where $\mathbf{l} = (l_1, l_2, \ldots, l_N)^T$ is a set of binary labels for all nodes in the graph. D_i is a data association function operating on each pixel, $V_{i,j}$ is an interaction potential and n is a set of neighbouring pixels or clique. In the system of [Boykov & Jolly (2001)] a user supplies a set of hard constraints as background and foreground pixels through broad brush-strokes. They become associated to source or sink nodes. These seed pixels can also be used to estimate foreground and background region statistics. The capacities or weightings of pixels can then be derived from region and boundary terms. The max-flow/min-cut problem is then solved and the pixels are assigned labels according to the source or sink they remain connected to. The basic system of [Boykov & Jolly (2001)] has been extended in further works. The GrabCut system [Rother *et al.* (2004)] iteratively re-estimates region statistics, which are modeled as mixtures of Gaussians, hence enabling the system to opperate with reduced user interaction. [Cui *et al.* (2008)] use colour and edge models from prior segmentations to improve local models used in the GrabCut framework. Other work investigates the addition of shape priors to use knowledge about an objects shape during the segmentation processes [Lempitsky & Boykov (2007),Lempitsky *et al.* (2008)]. An experimental comparison for minimising the energy of equation 6.20 can be found in [Boykov & Komolgorv (2004)].

The above approach to optimising graph cut energy functionals requires the use of combinatorial optimisation techniques. By relaxing the binary energy of equation 6.20 to a $[0, 1]$ random field results in a quadratic energy functional which can be solved by standard techniques from linear algebra. Such relaxations allow a probabilistic interpretation of the membership of image pixels within a class. A representative method is the random walker algorithm [Grady (2006)]. In follow up works [Couprie *et al.* (2009)] relate the random walker algorithm to watershed cuts and also to the Mumford-Shah functional in [Grady & Alvino (2009)].

3.6 Diffusion Processes in Image Processing: Gradient Vector Flow

Here to illustrate the challenges associated with the application of partial differential equation based diffusion processes in image processing we provide some experiments on the gradient vector flow problem. As discussed, gradient vector flow has been traditionally used in the literature as an external energy for forcing active contour models on image boundaries. That being understood it has more recently found applications in areas of image skeletonisation [Hassouna & Farag (2009)]. To briefly recap, traditionally the external energy driving active contour models was based only on edge information, derived from the image gradients. Let as write the gradient vectors of the edge map as

$$\nabla f = \nabla |\nabla G_\sigma * f|^2 \tag{3.16}$$

where notation is slightly abused. G is a Gaussian kernel with radius σ and f is the original image. Throughout this section we use ∇f to imply the gradient of the edge map as opposed to the gradient of the image. ∇f is known as the external potential or forcing function. This function pulls or drives the active contour towards image boundaries. This function only has large values in the vicinity of "strong" edges which we considered to be the global minima and small values in the vicinity of "weak" edges which are usually the local minima. In order to increase the capture range and minimise the influence of local minima the gradient vector flow was proposed. It diffuses the gradients of the strong edges throughout the image based upon a partial differential equation diffusion process.

GVF can be defined as the vector field $\mathbf{u(x)} = [u_1(x_1, x_2), u_2(x_1, x_2)]$ that minimizes the functional [Xu & Prince (1998)]

$$F_{ext}(\mathbf{u}) = \frac{1}{2} \int_0^1 \mu |\nabla u_i|^2 + p(\nabla f)|\mathbf{u} - \nabla f| d\Omega. \tag{3.17}$$

Where μ is an arbitrary constant balancing the contribution between the diffusion and regularisation terms and $p(\nabla f)$ is a function describing the local variation

of the image. This is the two dimensional GVF. The following Euler-Lagrange equations

$$\left.\begin{aligned} \mu\nabla^2 u_i - p(\nabla f)(u_i - f_{x_i}) = 0, \quad \text{in } \Omega \\ \tfrac{\partial u}{\partial \vec{n}} = 0, \quad \text{in } d\Omega \end{aligned}\right\} \text{ for } i = 1, 2 \tag{3.18}$$

are a minimizer for the functional. In the original work of Xu and Prince [Xu & Prince (1998)] the GVF is then defined as the equilibrium solution of

$$\frac{\partial u_i}{\partial t} = \nabla^2 u_i - p(\nabla f)(u_i - f_{x_i}) \tag{3.19}$$

respecting the prescribed boundary conditions. In mathematical physics this is a reaction diffusion system associated with the name Helmholtz. The first term

$$\nabla^2 u_i \tag{3.20}$$

seeks to diffuse the vector field through the medium. This diffusion process is linear and isotropic. The second term

$$p(\nabla f)(u_i - f_{x_i}) \tag{3.21}$$

is a reaction term which seeks to reconcile the gradient f_{x_i} to u_i. This term is chosen to be active in regions of high local variation such that in these regions there is a close fit and in regions of low variation the diffusion term dominates the Euler-Lagrange equation. This diffusion process therefore extends the vector fields within the image.
Finite difference discretisation leads to the problem of solving sparse large linear systems of equations; the components of the vector field being decoupled and solved separately:

$$L_i \mathbf{u}_i = \mathbf{f}_i. \tag{3.22}$$

In the original work [Xu & Prince (1998)] an explicit time finite difference time integration scheme is then used to simulate the GVF field. Termination of the algorithm was chosen to be equal to the number of pixels of the longest image dimension of the image matrix [Han *et al.* (2007)] and hence this termination

strategy is arbitrary. The scheme is known to be inefficient as the computational cost of this method grows with the size of the image. For an image of n × n pixels the matrix L is N × N where N = n × n. As can be seen as the size of the dataset is increased the convergence of the algorithm does not scale well. A more efficient algorithm is therefore required to simulate the GVF fields. Due to the large size of the problem direct methods such as Gaussian Elimination and LU decomposition are inappropriate. These algorithms do not maintain the sparse nature of the matrix hence having high memory costs and are computationally expensive. Efficient iterative methods are sought to simulate the GVF fields.

3.6.1 Finite Difference Discretization

In the system of elliptic partial differential equations (PDEs) of equation (3.18) we take $p = p(\nabla f) = (f_{x_1}^2 + f_{x_2}^2)$. We then have the system of decoupled PDEs

$$\mu\nabla^2 u_1 - (f_{x_1}^2 + f_{x_2}^2)(u_1 - f_{x_1}) = 0$$
$$\mu\nabla^2 u_2 - (f_{x_1}^2 + f_{x_2}^2)(u_2 - f_{x_2}) = 0$$

and taking $f_1 = (f_{x_1}^2 + f_{x_2}^2)f_{x_1}$, $f_2 = (f_{x_1}^2 + f_{x_2}^2)f_{x_2}$ we can write the equation in a simplified notation

$$\mu\nabla^2 u_1 - pu_1 = -f_1$$
$$\mu\nabla^2 u_2 - pu_2 = -f_2.$$

At this point we drop the subscripts $_1$and$_2$. Hence u refers equally to u_1 and u_2 likewise f refers equally to f_1 and f_2 and we can write the finite difference discretization

$$\frac{\mu(+u_{i,j-1} + u_{i-1,j} - 4u_{i,j} + u_{i+1,j} + u_{i,j+1})}{h^2} - p_{i,j}u_{i,j} = f_{i,j} \tag{3.23}$$

on a square grid with Neumann boundary conditions where $h = \frac{1}{\Delta x} = \frac{1}{\Delta y}$ is the spatial discretisation step size; we are assuming a square domain for ease and efficacy of presentation. This can be written as the four point stencil

$$\frac{1}{h^2}\begin{bmatrix} & +\mu & \\ +\mu & 4\mu + h^2 p_{i,j} & +\mu \\ & +\mu & \end{bmatrix}. \tag{3.24}$$

3.6.2 Operator Conditioning

The discretization of the GVF equations produces an elliptic PDE operator. The operator L has the form

$$L = \frac{1}{h^2}\begin{pmatrix} T_0 & -I & \dots & \dots & 0 \\ -I & T_1 & -I & \dots & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & -I & T_n \end{pmatrix}.$$

where

$$T_i = \begin{pmatrix} 4\mu - h^2 p_{i,0} & -\mu & -\mu & \dots & 0 \\ -\mu & 4\mu - h^2 p_{i,1} & -\mu & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & -\mu & -\mu & 4\mu - h^2 p_{i,N} \end{pmatrix}.$$

and I is the identity matrix multiplied by μ. One can notice immediately that the diagonal elements of T_i are not constant. It should also be noted that the coefficients $h^2 d_i$ are bounded between 0 and 1. The condition number $\kappa(L) = \frac{\lambda_{max}(L)}{\lambda_{min}(L)}$ of this operator matrix increases as the size of the matrix increases; here $\lambda_{max}(L)$ and $\lambda_{min}(L)$ are the maximum and minmum eigenvalues of L. It is well understood that convergence of the conjugate gradient algorithm [Chan & Jin (2007)] is given by

$$\rho = \frac{2c(\kappa)}{1 + c^2(\kappa)} \tag{3.25}$$

where

$$c(\kappa) = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}. \tag{3.26}$$

Therefore the conjugate gradient algorithm converges faster when the operator matrix has a lower condition number.

3.6.3 Explicit Scheme - Operator Splitting Strategy

After finite difference discretization of the GVF equations we require a numerical scheme for solving or simulating the resultant matrix system.. The simplest scheme is an explicit time stepping scheme, as used in the original work on gradient vector flow [Xu & Prince (1998)]. The explicit scheme can be written in matrix form as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t(L\mathbf{u}^n - \mathbf{f}) \tag{3.27}$$

where Δt is a discrete time stepping parameter. Such a scheme is often referred to as Euler's integration and is explicit as the solution at iteration $n+1$ is fully determined explicitly as the solution at iteration n. The time stepping size, Δt is restricted in magnitude by the Courant-Friedrich-Lewy (CFL) condition [Morton & Mayers (2005)] to insure that the scheme is stable. This condition imposes the restriction that

$$\Delta t \leq \frac{h^2}{4\mu} \tag{3.28}$$

which implies that the time step for diffusing the gradient information through the image is proportional to the square of the resolution of the spatial discretization. Therefore the finer the discretization the smaller the time step restriction. This condition implies that the smaller time steps should be used for larger images. The numerical scheme therefore does not scale well with image size. We can make improvements on this scheme. A first improvement to this scheme can be effected by using an operator splitting strategy.

In the computational physics community operator splitting strategies are common for solving initial value problems. Operator splitting is a form of divide and conquer strategy on the operator level. The motivation of an operator split is to numerically integrate complicated problems with the limited computational resources. We notice that we can split the operator L such that we have a matrix A with Toeplitz structure and D to be a diagonal matrix. The split is additive such that

$$L = A + D.$$

This results in the explicit scheme

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t(A\mathbf{u}^n + D\mathbf{u}^n - f). \tag{3.29}$$

As A has a Toeplitz structure it is recognised from the literature [Strang (1986b), Strang (1986a), Ng (1995), Chan & Ng (1996), Chan & Ng (1996), Strang (1999), Lin (2001), Chan & Jin (2007)] that it can be efficiently multiplied with the vector $\mathbf{u}$ in the Fourier domain. To perform the matrix vector multiplication

$$A\mathbf{u} \tag{3.30}$$

we embed the matrix A in a $(2N-1) \times (2N-1)$ matrix C and perform the matrix vector multiplication as follows

$$\begin{pmatrix} A & A_c \\ A_c & A \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} A\mathbf{u} \\ A_c\mathbf{u} \end{pmatrix}$$

where A_c is a matrix that ensures the matrix C is circulant. Circulant matrices allow the decomposition

$$C = F^*\Lambda F \tag{3.31}$$

where F is the Fourier matrix and F^* is it's Hermitian transpose. The Fourier transform of a circulant matrix is fully determined by the first row (or column) of the circulant matrix [Strang (1986b)]. As a result we only need to store the values of the first row of the matrix A.

We also notice that the matrix D is diagonal hence we can store this matrix as an $N \times 1$ vector, say $\mathbf{d}$. In order to perform the multiplication

$$D\mathbf{u} \tag{3.32}$$

we can perform an element wise scalar multiplication of the vector elements of $\mathbf{d}$ and $\mathbf{u}$.

The proposed operator splitting strategy reduces the storage requirements of the explicit time difference strategy proposed in [Xu & Prince (1998)]. Instead of storing a sparse matrix L of size $N \times N$ we instead store two sparse vectors of size $N \times 1$. However the algorithm does not scale well as Δt is constrained by the CFL condition. To further improve performance we therefore propose an implicit scheme.

3.6.4 Implicit Scheme - Operator Splitting Strategy

An implicit time stepping scheme can be written in the form

$$\mathbf{u}^{n+1} - \Delta t A\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t(D\mathbf{u}^n - \mathbf{f}). \tag{3.33}$$

In this scheme we have again used the operator splitting $L = A + D$. This scheme can be written in a simplified form as

$$(I - \Delta t A)\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t(D\mathbf{u}^n - \mathbf{f}). \tag{3.34}$$

This system requires the inversion of the matrix

$$M = (I - \Delta t A) \tag{3.35}$$

at each time step. The advantage of this operator splitting scheme over the explicit scheme is that it is no longer conditionally stable on the time stepping parameter Δt. As a result we can increase the magnitude of this time stepping parameter and convergence would still be guaranteed in a small number of iterations. Secondly such a scheme should scale better with image size.
We notice that the matrix M has a Toeplitz structure. This is advantageous as it is well understood from the literature [Strang (1986b), Strang (1986a), Ng (1995), Chan & Ng (1996), Chan & Ng (1996), Strang (1999), Lin (2001), Chan & Jin (2007)] that systems involving Toeplitz matrices can be solved efficiently in the Fourier domain. We again embed the matrix M into a circulant matrix

$$C = \begin{pmatrix} M & M_c \\ M_c & M \end{pmatrix} \tag{3.36}$$

where M_c is used to ensure that C is circulant. Inversion of circulant matrices can be effected by the following decomposition

$$C^{-1} = F^* \Lambda^{-1} F \tag{3.37}$$

where Λ is the matrix of eigenvalues of C. This is a diagonal matrix with the eigenvalues running along the main diagonal. These eigenvalues are obtained efficiently with computational complexity $\mathrm{O}(nlog(n))$ operations by taking the

FFT of the first row (or columns) of C.
We can therefore run two variants of this operator splitting strategy which we will call GVF-AOS1 and GVF-AOS2. GVF-AOS1 directly implements equation 3.34. This can be written algorithmically as in algorithm 2 The matrix system

Algorithm 2 GVF-AOS1

choose Δt, μ
initialise $\mathbf{u}^0$
for $n = 0, 1, 2, \ldots$ until convergence **do**
    $\mathbf{b}^n = \mathbf{u}^n + \Delta t(D\mathbf{u}^n - \mathbf{f})$
    solve $M\mathbf{u}^{n+1} = \mathbf{b}^n$
end for

involving M can be solved using a standard method from computational linear algebra, such as the conjugate gradient method or multigrid methods.
A second strategy is to use the circulant approximation of the matrix M as a preconditioning (or prediction) step. A second step is then used to refine the solution. This can be represented algorithmically as in algorithm 3 where FFT is the fast Fourier transform and iFFT its inverse. We call this strategy GVF-AOS2. We can explicitly construct the matrix C by introducing -1 to the elements

Algorithm 3 GVF-AOS2

choose Δt, μ
initialise $\mathbf{u}^0$
assemble C
compute $\Lambda = \mathrm{FTT}(C)$
for $n = 0, 1, 2, \ldots$ until convergence **do**
    $\mathbf{b}^n = \mathbf{u}^n + \Delta t(D\mathbf{u}^n - \mathbf{f})$
    $\hat{\mathbf{b}}^n = \mathrm{FFT}(\mathbf{b}^n)$
    $\hat{\mathbf{u}}^n = \mathrm{iFFT}(\Lambda^{-1}\hat{\mathbf{b}}^n)$
    solve $M\mathbf{u}^{n+1} = \hat{\mathbf{u}}^n$
end for

$M_{1,N}$ and $M_{N,1}$ of the matrix M. This would ensure that C meets the circulant requirement.

3.7 Numerical Experiments

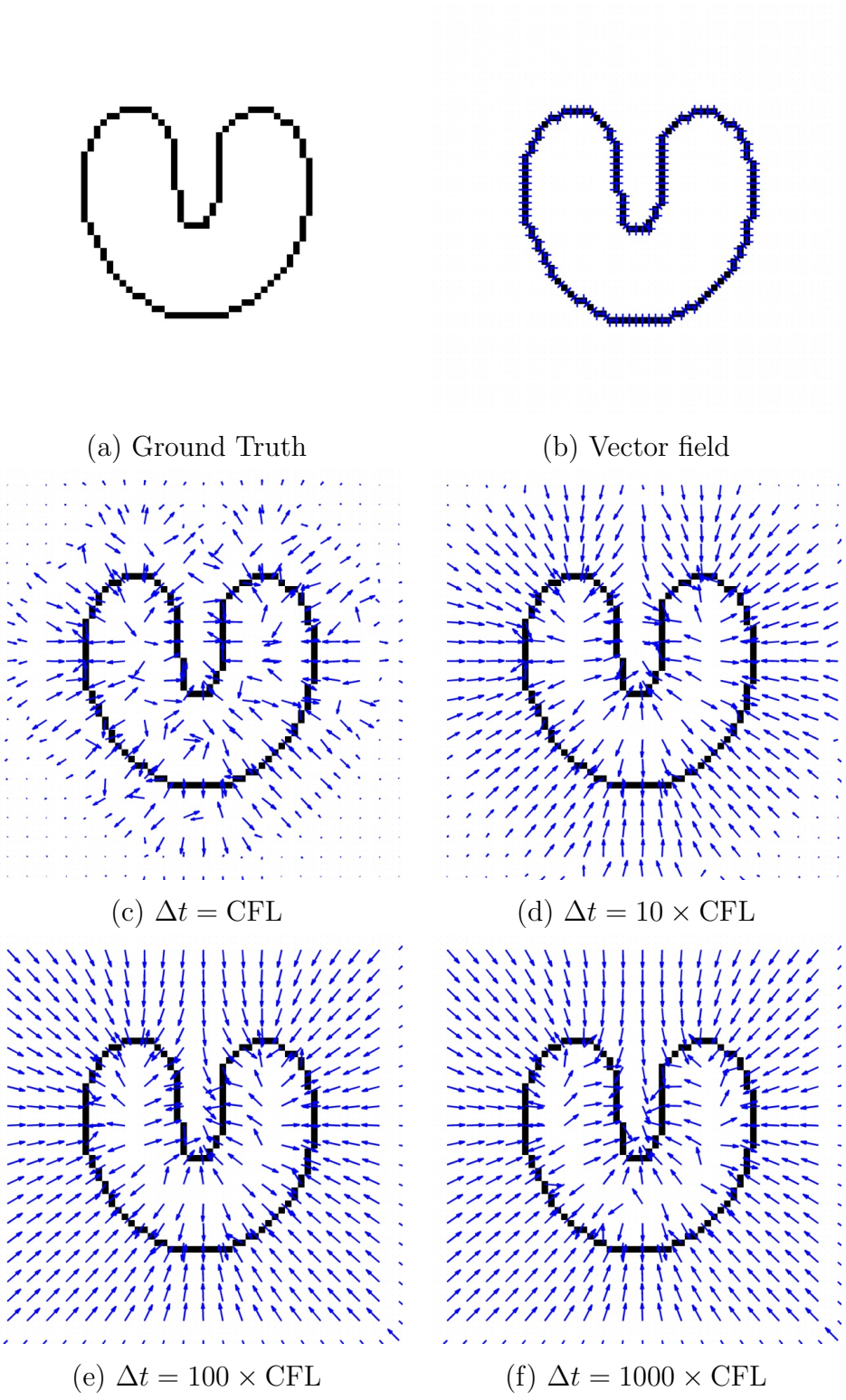
The first experiment investigates the sensitivity of the explicit schemes to the parameter Δt. To investigate this effect we use the synthetic image of figure 3.1 (a). This figure contains a well defined binary image. This image is useful as we expect the image gradient vectors to point in the direction of the black edge as in figure 3.1 (b), which is the initial gradient vector. We then vary the parameter Δt. We choose Δt = CFL number, $\Delta t = 10 \times$ CFL number, $\Delta t = 100 \times$ CFL number and $\Delta t = 1000 \times$ CFL number. We then view the gradient vector fields at the convergence of the algorithm, figures 3.1 (c), (d), (e) and (f). We choose convergence as the condition when $\epsilon < 1 \times 10^{-7}$ is satisfied. The parameter ϵ is taken to be the residual

$$\epsilon = \|\mathbf{u}^{n+1} - \mathbf{u}^{n}\|_2^2. \tag{3.38}$$

From this experiment we see that for a choice of Δt = CFL we do not recover a good GVF feld. The recovered field is improved as we increase $\Delta t = 1000$. In fact the scheme is still stable at $\Delta t = 1000 \times$ CFL. The poor GVF field for Δt = CFL could be understood as being due to the dominance of the identity matrix. Recall that

$$M = I - \Delta t A. \tag{3.39}$$

For low values of Δt this tends to the identity matrix.
In the second experiment we compare the computational speeds of the different numerical schemes. For this experiment we use the liver CT-scan of figure 3.2 (a). We scale the image to the sizes 63×63, 127×127 and 255×255. We then run the computational experiments with the explicit scheme , the explicit scheme with FFT multiplication, GVF-AOS1 and GVF-AOS2. For all schemes we fix $\mu = 1.5$. For the explicit schemes Δt = CFL number and for the implicit schemes (GVF-AOS1 and GVF-AOS2) we fix $\Delta t = 1000 \times$ CFL number. All experiments are run 100 times with the average run time reported. The experiments are run in MATLAB with no C/C++ optimisations. We run the experiments on an AMD Athalon $64 \times$ X2. Finally we compare with the multigrid method proposed in [Han *et al.* (2007)]. Multigrid strategies are iterative numerical solvers that solve

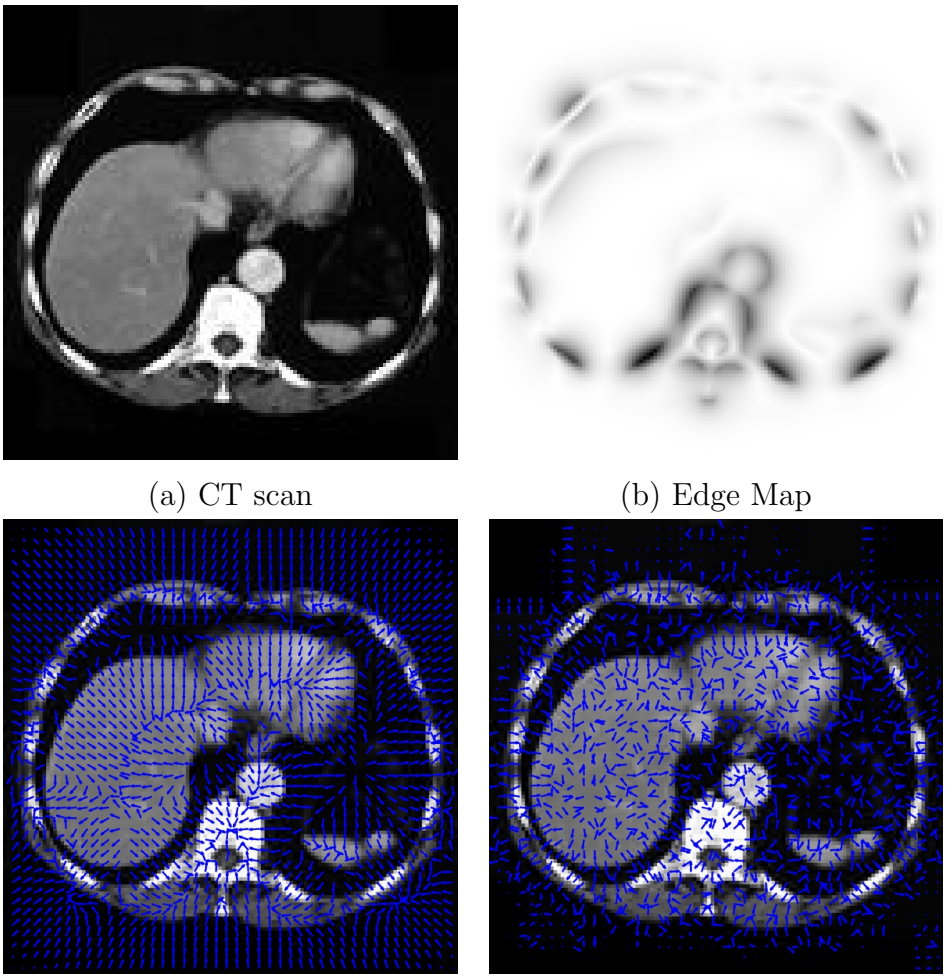


(a) Ground Truth (b) Vector field

(c) $\Delta t = \text{CFL}$ (d) $\Delta t = 10 \times \text{CFL}$

(e) $\Delta t = 100 \times \text{CFL}$ (f) $\Delta t = 1000 \times \text{CFL}$

Figure 3.1: Simulation of the GVF fields using an implicit time stepping scheme with varying values of Δt. The parameter μ is set to 1.5 in all experiments.



(a) CT scan (b) Edge Map

(c) GVF (Neuman) (d) GVF (reflective)

Figure 3.2: Simulation of GVF fields on a liver CT-scan. Parameters $\mu = 1.5$ and $\Delta t = 1000 \times CFL$ using the GVF-AOS1 numerical scheme. (a) CT scan, (b) edge map, (c) simulation with Neumann boundary conditions and (d) is simulation with reflective boundary conditions.

linear equations by constructing a sequence of progressively coarser grids and applying the idea of nested iteration and coarse grid refinement to accelerate the solution process of iterative methods such as the Jacobi method, weighted Jacobi and related methods [Saad (2003)]. Full details of multigrid methods can be found in standard texts such as [Briggs *et al.* (2000), Saad (2003)]. As suggested in [Han *et al.* (2007)] we use the full multigrid strategy (FMG) and experiment with the pre and post smoothing parameters.

We can notice from figure 3.2 (c) and (d) that the choice of the boundary

Algorithm	Max residual	Time	Speed Gain	Iterations
Explicit	6.96×10^{-7}	1.80	1	1000
GVF-AOS1	6.29×10^{-8}	0.53	3	3
GVF-AOS2	6.29×10^{-12}	0.56	3	3
FMG(6,7)	7.54×10^{-8}	0.28	6	

Table 3.1: Comparison of numerical schemes for simulation of GVF fields on the liver CT-scan size 63×63.

Algorithm	Max residual	Time	Speed Gain	Iterations
Explicit	5.52×10^{-7}	6.73	1	1000
GVF-AOS1	1.10×10^{-8}	2.02	3	3
GVF-AOS2	6.21×10^{-14}	2.03	3	3
FMG(5,6)	1.59×10^{-8}	0.92	7	

Table 3.2: Comparison of numerical schemes for simulation of GVF fields on the liver CT-scan size 127×127.

conditions affects the solution of the GVF field. It is well established in the literature of partial differential equations [Morton & Mayers (2005)] that numerical simulation of partial differential equations requires a "good" choice of the boundary conditions. We can see that this too is the case when generating the GVF fields.

From the computational experiments reported in tables 3.1, 3.2 and 3.3 we can see

Algorithm	Max residual	Time	Speed Gain	Iterations
Explicit	2.16×10^{-7}	36.20	1	1000
GVF-AOS1	8.43×10^{-10}	8.05	4	3
GVF-AOS2	9.79×10^{-16}	8.53	4	3
FMG(3,4)	7.22×10^{-8}	3.39	11	

Table 3.3: Comparison of numerical schemes for simulation of GVF fields on the liver CT-scan size 255×255.

that the implicit schemes have greatly improved on the computational speed over the original method (explicit) for numerical simulation of the gradient vector flow fields. In fact at termination of the GVF-AOS algorithms we can observe that they have achieved a very low tolerance with GVF-AOS2 consistently achieving the lowest tolerance. Comparisons with the multigrid method reported in [Han *et al.* (2007)] reveals that the full multigrid strategy (FMG) outperforms the algorithms GVF-AOS1 and GVF-AOS2 by an order of magnitude for all image sizes. A disadvantage of the FMG strategy proposed in [Han *et al.* (2007)] is that we have to perform a parameter search for the pre and post smoothing parameters that achieve the desired tolerance of the residual. This can be seen in tables 3.4, 3.5 and 3.6 where we have performed a manual parameter search of the FMG pre and post smoothing parameters in order for the solver to achieve a residual tolerance of $< 1 \times 10^{-7}$.

Algorithm	Residual	Residual	Time
FMG(1,1)	9.22×10^{-6}	8.90×10^{-6}	0.203
FMG(1,2)	5.20×10^{-6}	4.71×10^{-6}	0.204
FMG(2,1)	4.93×10^{-6}	4.93×10^{-6}	0.218
FMG(2,2)	3.00×10^{-6}	2.53×10^{-6}	0.219
FMG(2,3)	1.90×10^{-6}	1.58×10^{-6}	0.218
FMG(3,2)	1.93×10^{-6}	1.53×10^{-6}	0.219
FMG(3,3)	1.25×10^{-6}	9.72×10^{-7}	0.219
FMG(3,4)	8.04×10^{-7}	6.34×10^{-7}	0.234
FMG(4,3)	8.67×10^{-7}	8.51×10^{-7}	0.234
FMG(4,4)	5.61×10^{-7}	4.27×10^{-7}	0.235
FMG(4,5)	3.55×10^{-7}	2.83×10^{-7}	0.235
FMG(5,4)	4.08×10^{-7}	3.06×10^{-7}	0.235
FMG(5,5)	2.59×10^{-7}	2.03×10^{-7}	0.250
FMG(5,6)	1.59×10^{-7}	1.37×10^{-7}	0.265
FMG(6,5)	1.95×10^{-7}	1.53×10^{-7}	0.265
FMG(6,6)	1.21×10^{-7}	1.04×10^{-7}	0.266
FMG(6,7)	7.23×10^{-8}	7.54×10^{-8}	0.281

Table 3.4: Parameter search for pre and post smoothing parameter of FMG algorithm on 63×63 CT-scan image.

3.8 Discussion

Diffusion Processes in Image Processing. In this chapter we provided further discussions on diffusion processes in image processing. We focussed on the gradient vector flow problem which utilises a linear isotropic diffusion process. Our aim was to discuss the issues of applying partial differential equation based diffusion processes in image processing with emphasis on

- discretisation errors
- approximation of boundary conditions
- computational efficiency and stability problems.

Algorithm	Residual	Residual	Time
FMG(1,1)	4.20×10^{-6}	4.20×10^{-6}	0.672
FMG(1,2)	2.26×10^{-6}	2.33×10^{-6}	0.672
FMG(2,1)	2.09×10^{-6}	2.00×10^{-6}	0.672
FMG(2,2)	1.14×10^{-6}	1.21×10^{-6}	0.734
FMG(2,3)	7.02×10^{-7}	7.56×10^{-7}	0.734
FMG(3,2)	6.60×10^{-7}	7.34×10^{-7}	0.765
FMG(3,3)	4.14×10^{-7}	4.68×10^{-7}	0.781
FMG(3,4)	2.72×10^{-7}	3.07×10^{-7}	0.797
FMG(4,3)	2.70×10^{-7}	3.20×10^{-7}	0.828
FMG(4,4)	1.78×10^{-7}	2.10×10^{-7}	0.829
FMG(4,5)	1.22×10^{-7}	1.38×10^{-7}	0.844
FMG(5,4)	1.26×10^{-7}	1.54×10^{-7}	0.859
FMG(5,5)	8.67×10^{-8}	1.01×10^{-7}	0.890
FMG(5,6)	1.59×10^{-8}	1.37×10^{-8}	0.922

Table 3.5: Parameter search for pre and post smoothing parameter of FMG algorithm on 127×127 CT-scan image.

In the discretisation of the diffusion process we used a simple standard finite difference approximation. The finite difference operators are derived from Taylor series expansions. Taylor series expansions are infinite series therefore the approximations are truncated at low order terms. Even though we did not experimentally investigate the truncation errors we understand that the finite difference schemes are associated with such errors. Secondly from the experiments we performed on the gradient vector flow simulation we saw that choice of the boundary conditions influences the recovered solution. Unfortunately boundary conditions are necessary as we have to approximate the behaviour of the partial differential equation at the boundary of the finite image domain where as an ideal partial differential equation model would be formulated on an ideal infinite domain. Finally we saw that there is a trade of between the stability of the numerical scheme and the computational speed. This, in the partial differential equation and fludi dynamics literature is encapsulated in the CFL number. For our problems we

Algorithm	Residual	Residual	Time
FMG(1,1)	1.09×10^{-6}	1.05×10^{-6}	2.78
FMG(1,2)	5.78×10^{-7}	5.58×10^{-7}	2.93
FMG(2,1)	5.10×10^{-7}	5.03×10^{-7}	2.98
FMG(2,2)	2.88×10^{-7}	2.88×10^{-7}	3.02
FMG(2,3)	1.77×10^{-7}	1.78×10^{-7}	3.03
FMG(3,2)	1.67×10^{-7}	1.74×10^{-7}	3.12
FMG(3,3)	1.05×10^{-7}	1.12×10^{-7}	3.13
FMG(3,4)	6.87×10^{-8}	7.22×10^{-8}	3.39

Table 3.6: Parameter search for pre and post smoothing parameter of FMG algorithm on 255×255 CT-scan image.

were able to use simple operator split strategies to minimise the effect of the CFL number. Balancing this trade off becomes more difficult when more complex diffusion processes are introduced into the image processing problem such as those reviewed in the chapter prior and those used in level set implementations. As a result implementation of partial differential equation based diffusion processes often require many implementation choices such as boundary conditions, time stepping parameters, discretisation method and of efficient numerical solvers. As we have seen some of the choices may even produce differences in results.

Energy-based methods. As stated we could say that the low-level image segmentation problem is that of

finding groups of pixels that "go together".

A common idea that is shared by active contour methods and graph cut based methods is that they can be formulated as pixel-based energy functionals:

$$E = E_{data} + E_{smooth} \tag{3.40}$$

where the data term enforces that a function, feature or pixel label is similar to the data based upon gradients, colour or statistical information. The smoothness term imposes constraints on the smoothness or similarity of pixels in a region or boundary. When this problem is formulated in the continuous sense it leads to a

variational formulation which can be optimised by an active contour strategy. We have seen from the literature that active contour models were initially developed using a data term that matched image gradients. The image gradients are a forcing function that drive the active contour by a diffusion process. This strategy has difficulties as the active contour can get trapped in local minima. As a result forcing functions such as the gradient vector flow were introduced. Further to this the energy functional was augmented with region statistics an interactive inputs. Region statistics can be as simple as gray level intensities [Chan & Vese (2001)] or as complex as statistics derived from colour, texture and motion [Cremers *et al.* (2007)].

In the discrete setting optimising the pixel-based energy functional 4.22 leads to a discrete labeling problem. To tackle such problems binary optimsation techniques [Greig *et al.* (1989)] and min-cut/max-flow algorithms were introduced to effect the optimisation. In this framework a user labels a small subset of pixels as either foreground or background using broad brush-strokes [Boykov & Jolly (2001)] or a bounding box [Rother *et al.* (2004)] . A combinatorial optimisation algorithm then finds the labeling for the remaining pixels. Such strategies have been shown to overcome the local minima problems associated with the active contour formulations. In this chapter we also reviewed the normalised cuts algorithm. The method is formulated on a graph, like the graph cuts methods, but uses a continuous optimisation framework. It is a spectral clustering method and should not be confused with the graph cut strategies.

The presented segmentation algorithms are by no means an exhaustive representations of the methods available in the literature. In this chapter we only touched on methods and models that directly influence the work in this thesis.

Chapter 4

Discrete Regularisation on Weighted Graphs

4.1 Chapter Summary and Contributions

In this chapter we study diffusion processes on graphs as applied to the image denoising problem. Although the problem of image denoising is already well established and studied within the literature we tackle the problem from a new direction: through a graph theoretic approach. We review the recently proposed graph-based differential geometric arguments and recover the key innovations. Building upon these ideas we propose an alternate approach using graphical models and Markov random fields from which we derive a novel image denoising algorithm. We link the algorithm with the de facto energy minimisation methods in computer vision using the Markov random field. Moreover we interpret the new algorithm using nonparametric statistics and kernel density estimation.

The proposed algorithm is mathematically equivalent to the Google PageRank algorithm [Brin & Page (1998),Page *et al.* (1999)] for ranking hypertextual world wide web documents where by the associated Markov process is generated from the photometric similarities between image pixels. This algorithm has well define and understood numerical properties. In effect our derivation using Markov random fields is a novel alternate derivation of the Google PageRank algorithm. Such a derivation is not available in the literature.

Our approach continues the idea developed in this thesis of forming a diffusion

process on a graph and allowing information to diffuse through the nodes in order to effect image manipulation tasks. We also discuss the relationship between our algorithm and the non-local means algorithm Buades *et al.* (2005a). Finally due to the distribution of the eignvalues of the Markov process we propose an efficient solution using the conjugate gradient algorithm of Hestenes and Stiefel [Hestenes & Stiefel (1952),Golub & Van Loan (1996)].

4.2 Graph Based Image Denoising

We begin with a quick recap of calculus of variation and diffusion models for image processing. Here a typical formulation is to assume the noise is generated by a zero mean Gaussian processes allowing the decomposition

$$f = u + \eta, \tag{4.1}$$

where f is the observed signal, u is the original signal and η is a zero mean Gaussian process with finite variance. Variational image processing algorithms typically seek to minimize two energy terms; that is a data fidelity term and a discrete weighted regularizing term which is also know as a p-Dirichilet form. In the continuous scalar setting the problem is that of minimising for u the form

$$\frac{1}{p}\int |\nabla u|^p d\Omega + \frac{\lambda}{2}\int (u-f)^2 d\Omega$$

where $p \in (0, +\infty)$. When $p = 2$ the Tikhonov inverse problem is recovered and when $p = 1$ we recover the total variation minimization problem. After optimising this problem we are faced with casting the optimiser into a discrete form. Literature on these nonlinear partial differential equation diffusion filters can be found in [Perona & Malik (1990b),Weickert *et al.* (1998),Chambolle (2004a),Bresson & F (2008)]. These algorithms have been largely successful but can be computationally expensive [Weickert *et al.* (1998)]. As a result elaborate numerical linear algebra techniques using conjugate gradient and multigrid algorithms have been proposed to solve these systems [Vogel & Oman (1996),Weickert *et al.* (1998),Chambolle (2004b),Strümer *et al.* (2008),Aujol (2009)]. Although having a succinct mathematical formulation these methods have not been as competitive as denoising methods using *transform-shrinkage-inverse transform* strategies.

4.2.1 Overview

Graph-based regularisation. Related filters to the diffusion filters have been developed that seek to average image intensities by means of a nonlinear combination of nearby image values. Similarity and weighting of the pixels is based upon geometric closeness and photometric similarity. Such filters are exemplified by the bilateral filter [Tomasi & Manduchi (1998)] and the non-local means filter [Buades *et al.* (2005a),Buades *et al.* (2005b)].
Recently image denoising algorithms have been developed that diffuse image intensities based upon models that capture the topological structure of images. The topological structure of the image is modeled by photometric and geometric similarities within the image. The algorithms are based upon graph formulations. Algorithms based upon graph formulations have appeared in the form of normalized cuts [Shi & Malik (2000)] for image segmentation, heat kernel smoothing for image denoising [Zhang & Hancock (2006)] and random walks and Markov processes for image denoising [Azzabou *et al.* (2006),Estrada & Jepson (2009)]. A common idea these methods share is that a priori beliefs about the correlations or similarities between pixels are captured in a kernel matrix. This matrix reflects the graph adjacency structure and can be studied by forming the graph Laplacian [Chung (1997)]. They exploit the assumption that

> *pixels in local neighbourhoods are self-similar.*

These filters can be seen to be weighted neighbourhood filters with a strong relationship to PDE diffusion filters [Bougleux *et al.* (2009),Gilboa & Osher (2007)]. As the image data is inherently discrete the regularisation problem is formulated directly in the discrete setting. A graph based model of the image is used to represent the image data. The local and nonlocal pixel similarities can be captured and modeled in the regularisation framework.
Graph based regularisation formulations have been introduced in a semi-supervised problems in machine learning [Zhou *et al.* (2003), Zhou & Schlkopf (2004), Zhou & Schlkopf (2005)] and formalized for image, mesh and manifold processing in [Elmoataz *et al.* (2008), Bougleux *et al.* (2009)]. In general these methods seek

to minimise the signal vector for $\mathbf{u}$ over the regularised functional

$$\frac{1}{p}\sum_{x\in V}\|\nabla_x \mathbf{u}\|^p + \frac{\lambda}{2}\|\mathbf{u}-\mathbf{f}\|^2 \tag{4.2}$$

defined on a graph $G = (V, E, w)$ consisting of a set of vertices V, a set of edges $E \subset V \times V$, a similarity function w defined on the edges and $\|\nabla_x \mathbf{u}\|$ is the local variation of the function u at a vertex x on the graph. G is assumed symmetric with no self loops or multiple edges and $u : V \to \mathbb{R}_+$ is a real valued function assigning a value to each vertex $u(x)$. It is in the Hilbert space of real functions $\mathbb{H}(V)$ defined on the set of vertices. The solution to this funcitonal leads to a family of nonlinear processing methods which are determined by the weight function, the degree of smoothness and the choice of fidelity parameter λ. It has been shown [Zhou *et al.* (2003),Elmoataz *et al.* (2008)] that the solution to this regularization problem is of the form

$$\Delta_p u + \lambda(u - f) = 0 \tag{4.3}$$

where Δ_p is a Laplace-Beltrami operator also known as the p-Laplacian operator. The minimisation can be considered as a discrete analogue of the Euler-Lagrange equations defined on a graph and is a discrete diffusion process on a graph.

Graph-based semi-supervised learning. We would like to take advantage of the geometrical structure inherent within the signals in order to develop our filtering algorithms. The initial difficulty is to capture the geometric structure within the signal. These structures include discontinuities and rapid transitions. A graph formulated upon the signal allows us to reveal this structure. As such our signal processing problem has relations to mesh smoothing and processing and graph based problems in transductive inference and semi-supervised machine learning [Zhou & Schlkopf (2004)]. For such problems we are given a domain for the data

$$\{x_1, x_2, \ldots, x_N\} \tag{4.4}$$

with a set of associated outputs or labels for a subset of the data

$$\{f(x_1), f(x_2), \ldots, f(x_i)\}. \tag{4.5}$$

We may know the links and relationships between the data, a priori, and the problem is then to determine the remaining labels or outputs

$$\{f(x_i), \ldots, f(x_N)\}. \tag{4.6}$$

Transductive inference has an intimate relationship to semi-supervised learning and as such the problems are studied in a similar framework. They have been applied to problems such as web-page ranking, co-authorship networks and biological networks [Kondor & Lafferty (2002),Zhou & Schlkopf (2004)]. Under this framework the nodes of the graph represent a datum and the edges encode a pairwise similarity between the data [Kondor & Lafferty (2002),Zhou & Schlkopf (2004)]. The similarities between the data are often encoded using a Gaussian random field model [Zhu *et al.* (2003a)], that is the node similarities are encoded using a normalised exponential measure. This connection to semi-suervised machine learning algorithms shall be further exploited to derive image segmentation algorithms.

Important contributions to graph regularisation. An important contribution to the problem of diffusion of information through the graph came by the development of a regularisation framework for learning from graph data [Zhou & Schlkopf (2004)]. In this paper a graph derivative operator and regularisation framework on graphs was proposed and developed and related to lazy random walks on graphs. Further developments to this framework were proposed in [Zhou *et al.* (2004b), Zhou *et al.* (2004a)]. The framework introduces the Hilbert spaces on vertices and edges on a graph $\mathbb{H}(V)$ and $\mathbb{H}(E)$ respectively. A difference operator, d, which is a mapping from the Hilbert space of edges is introduced. The adjoint of this operator which is a mapping from the Hilbert space of edges to the Hilbert space of vertices is defined. These operators are used to define the discrete Laplace-Beltrami operator on the graph. This approach was introduce in machine learning in [Zhou & Schlkopf (2004)]. In [Zhou *et al.* (2004b)], the diffusion model was applied to web page ranking, in particular to the PageRank algorithm [Brin & Page (1998),Page *et al.* (1999)]. Practical methods tend to use an iterative update [Hein *et al.* (2005),Zhou & Schlkopf (2004),Nadler *et al.* (2005)].

This diffusion framework requires the formulation of the graph Laplacian, although successful approaches in image processing have been based upon spectral decomposition [Zhang & Hancock (2006)]. It is generally-believed in the machine learning community that the discrete graph Laplacians converge to the continuous Laplace-Beltrami operator when the number of discrete samples tends to infinity [Hein *et al.* (2005)]. As the Laplace-Beltrami operator on a manifold can be seen to be a generator of a diffusion process [Nadler *et al.* (2005)] the discrete operator can be thought of as a generator of data on the vertices of the graph. Some convergence results of the discrete operator are presented in [Hein *et al.* (2005)].

In the particular choice of the normalised graph Laplacian we have a Markov matrix. Under this interpretation we have a powerful framework for data analysis. Applications of such a framework have been studied in data classification and clustering [Coifman & Lafon (2006)] proving efficient for finding relevant structures in complex nonlinear geometries. The L_1 distance between transition probabilities, for instance, has been used as a metric to induce class labels on labeled data [Szummer & Jaakkola (2002)]. Spectral clustering under this framework has been studied in [Meila & J (2001)]. The eigenvectors of the Markov matrix can be seen as a score for data points leading to ranking techniques. The PageRank algorithm [Brin & Page (1998), Page *et al.* (1999)] for web page ranking uses the stationary distribution of random walks induced by the Markov chain of the link structure of the web in order to rank web page documents in terms of relative importance. The top eigenvectors of the Markov matrix provide information for page ranking [Lempel & Moran (2000)].

Key concepts. The important concept from these approaches is generating the Laplacian matrix encoding the links and similarities inherent within the signal. We believe, from the literature, the preferred Laplacian should be a Markov matrix. Power iterations of the associated random walk forward in time allows us to relate the spectral properties of the diffusion process to the geometry of the signal. In the literature graph divergence, gradient and Laplacian operators have been defined. These are analogous to the continuous differential operators defined on a manifold. From such operators graph-based diffusion processes have been defined.

4.3 The Graph Regularization Framework

4.3.1 Functions, Gradients and Divergence Operators

Given a signal defined on each vertex of the graph $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T \in V$ we represent it as a function $f : V \mapsto \mathbb{R}^+$ on the vertices. This function implies the intensity of the signal at each vertex and we write

$$\mathbf{f} = (f(x_1), f(x_2), \ldots f(x_N))^T. \tag{4.7}$$

To analyse and process signals on the manifold we require the usual notions of a differential calculus. We seek to review gradient and divergence operators defined over spaces of vertices and edges of the graph G. The operators will allow us to recover a Laplace-Beltrami operator important for smoothing and denoising in image and mesh processing applications. It has been noted by many authours the difference in graph differential operators and the Laplace-Beltrami operators they imply [Chung (2007),Hein *et al.* (2005),Elmoataz *et al.* (2008),Bougleux *et al.* (2009)]. We motivate our operator by the implied probabilistic filter. If we let $\mathbb{H}(V)$ denote the Hilbert space of real valued functions defined on the vertices of the graph $f, g : V \to \mathbb{R}$ endowed with the usual inner product

$$< f, g >_{\mathbb{H}(V)} = \sum_{x_i \in V} f(x_i) g(x_i), \tag{4.8}$$

and $\mathbb{H}(E)$ denote the Hilbert space of real valued functions defined on the edges of the graph endowed with the inner product

$$< F, H >_{\mathbb{H}(E)} = \sum_{x_i \in V} \sum_{x_j \sim x_i} F(x_i, x_j) H(x_i, x_j). \tag{4.9}$$

where these functions need not be symmetric that is

$$H(x_i, x_j) = H(x_j, x_i) \tag{4.10}$$

is not always true. We denote by $x_j \sim x_i$ the edge from node x_j incident upon x_i and motivated by similar definitions of the edge derivative in [Zhou *et al.*

(2003),Elmoataz *et al.* (2008)] we define the edge derivative of a function at the node x_i as

$$(\partial f)([x_i, x_j]) = \sqrt{\frac{w(x_i, x_j)}{d(x_i)}}(f(x_j) - f(x_i)). \tag{4.11}$$

Our operator is a normalised version of the operator appearing in [Elmoataz *et al.* (2008)]. As a result we can define the weighted gradient operator as a mapping $\nabla : \mathbb{H}(V) \rightarrow \mathbb{H}(E)$ from the Hilbert space of functions on the vertices of the graph to the Hilbert space of functions of edges of the graph implying the vector

$$(\nabla f)(x_i) = \begin{pmatrix} (\partial f)([x_i, x_1]) \\ (\partial f)([x_i, x_2]) \\ \vdots \\ (\partial f)([x_i, x_{i-1}]) \\ (\partial f)([x_i, x_i]) \\ (\partial f)([x_i, x_{i+1}]) \\ \vdots \\ (\partial f)([x_i, x_{N-1}]) \\ (\partial f)([x_i, x_N]) \end{pmatrix}.$$

An adjoint difference operator can be defined such that

$$\partial^* : \mathbb{H}(E) \rightarrow \mathbb{H}(V) \tag{4.12}$$

with the property

$$< \partial f, H >_{\mathbb{H}(E)} = < f, \partial^* H >_{\mathbb{H}(V)} . \tag{4.13}$$

4.3.2 The Local Variation on a Graph

The derivative operator induces two important concepts on our signals. Firstly it introduces a probabilistic model to our framework. The normalised weightings between nodes

$$p(x_i|x_j) = \frac{w(x_i, x_j)}{d(x_i)} \tag{4.14}$$

can be interpreted as transition probabilities on the graph. Secondly this operator allows us to define the local variation of a signal at a node

$$\|(\nabla f)(x_i)\|_2 = \left(\sum_{x_j \sim x_i} (\partial f)^2([x_i, x_j]) \right)^{\frac{1}{2}}. \tag{4.15}$$

which is the L_2 norm over the weighted gradient at a node. This L_2 norm implicitly defines a filter through the local variation at a node. These filters have a close relationship to non-local means operator defined in [Buades *et al.* (2005a)]. Using this framework we can define the non local means at a node using

$$\begin{aligned} \mathrm{NL(f)}(x_i) &= \textstyle\sum_{x_j \sim x_i} \frac{w(x_i,x_j)}{d(x_i)} f(x_j) \\ &= \textstyle\sum_{x_j \sim x_i} p(x_i|x_j) f(x_j). \end{aligned}$$

which is the conditional expectation of the grey levels in a region. If we define the quadratic loss function as

$$L(f(x_j), f(x_i)) = (f(x_j) - f(x_i))^2 \tag{4.16}$$

then we can write

$$\|(\nabla f)(x_i)\|_2^2 = \sum_{x_j \sim x_i} p(x_i|x_j) L(f(x_j), f(x_i)). \tag{4.17}$$

The local variation of the signal at a node implies a filter which can be interpreted as the local expected loss of the signal in a neighbourhood. A filtering strategy may be to minimise the expected signal loss in a neighbourhood of pixels. This is intuitive as it follows from the notion that

pixels in local neighbourhoods are self-similar.

Minimising the expected signal loss can be seen as imposing this constraint.

4.3.3 Laplace Operator

If we define

$$\mathrm{div} = -\partial^* \tag{4.18}$$

then we can define the graph Laplacian as

$$(\Delta_{rw} f)(x_i) = \frac{1}{2\sqrt{d(x_i)}} \text{div} \sqrt{d}(\partial f) \tag{4.19}$$

From this expression we can derive the expression for the random walk Laplacian as

$$(\Delta_{rw} f)(u) = f(u) - \sum_{x_j \sim x_i} \frac{w(x_i, x_j)}{d(x_i)} f(x_j). \tag{4.20}$$

$$(\Delta_{rw} f)(x_i) = f(x_i) - \text{NL(f)}(x_i) \tag{4.21}$$

which is a normalised version of the graph Laplacian [Chung (1997)].

4.4 Markov Random Field Formulation

Before we dive into some mathematics let us recap on some ideas relating to our motivations. We recall hat we would like to use the de facto frameworks in computer vision in which to formulate our problems. In the high level view of such problems we use a pixel-based energy function

$$E = E_{data} + E_{smooth}. \tag{4.22}$$

From this viewpoint we would like to derive a probabilistic model that allows us to produce the edge preserving smoothing characteristic. Invariably from a probabilistic point of view we would like to understand our model. Firstly we consider deriving an algorithm using pairwise Markov random fields. We would particularly like to focus on the mechanism for optimisation. We know from the literature that generative Markov random field models are statistical models that seek to model the process that produces the observed data subject to our prior beliefs on the nature of the data. In essence a good model requires a good understanding of the statistics of the underlying process and the nature of the data.

Now we recall that the pixels of the image in a lattice structure may be represented as a graph $G = \{\mathbb{V}, \mathbb{E}\}$ where the pixels are the nodes $\{f_i \in \mathbb{V}; \forall f_i \in \mathbf{f}\}$ of the graph and $\mathbb{E}$ is the set of edges. If we assume $\mathbf{f}$ to be a Markov random field and considering K overlapping cliques then the probability density of the image can be expressed as the graphical model

$$p(\mathbf{f}) = \frac{1}{Z} \prod_{k=1}^{K} \Psi_k(\mathbf{f}_{(k)}) \tag{4.23}$$

which can be equivalently written as

$$p(\mathbf{f}) = \frac{1}{Z} \exp \left[- \sum_{k=1}^{K} U_k(\mathbf{f}_{(k)}) \right] \tag{4.24}$$

where $\mathbf{f}$ is the image, U_k is a potential function, Ψ_k, by an abuse of terminology, is also known as a potential function and $\mathbf{f}_{(k)}$ is a clique. A clique being a set of nodes of a graph where each node is a neighbour to the others. Each potential

function, U_k and Ψ_k, operates only upon the set of pixels in the clique. There is a normalizing term Z ensuring the Gibbs distribution integrates to 1. In this model we make the assumption and imposition that the MRF is homogenous; potential functions are the same for all cliques.

Higher order MRF models, such as the fields-of-experts [Roth & Black (2009)], model the potential functions Ψ_k as a product of expert distributions [Welling *et al.* (2002)]. This allows the model to be more expressive in capturing the distributions modeling the interactions of pixels.

What is assumed here is that we have a good model or understanding of the nature of the underlying data. The priors, for example, in the fields-of-experts models are modeled as a result of empirical studies on the natural statistics of images. These studies have shown that the marginal distributions of derivative filters of images are highly kurtotic or heavily peaked. The potential functions, therefore, modeling the prior information should be "expressive enough" to capture this behaviour. As such the potential function for each clique is modeled as

$$\Psi(\mathbf{f}_{(k)}) = \prod_{m=1}^{M} \phi(\mathbf{J}_m^T \mathbf{f}_{(k)}; \alpha_m) \tag{4.25}$$

where each $\mathbf{J}_m$ is a linear filter defined over the clique $\mathbf{f}_{(k)}$, ϕ is an expert probability distribution and α_m are the parameters of the expert distribution. We can see that this prior is a product of more primitive distributions which may be Gaussian, student t-distributions or even Gaussian scale mixtures [Schmidt *et al.* (2010)]. We have the freedom to choose the number of experts M dependent of the expressivity or the number of degrees of freedom required for each expert. In the case where $M = 1$ the potential function is just the expert probability distribution

$$\Psi(\mathbf{f}_{(k)}) = \phi(\mathbf{J}^T \mathbf{f}_{(k)}; \alpha). \tag{4.26}$$

For a general higher-order MRF the probability density function over the image can be written as

$$\begin{aligned} p(\mathbf{f}) &= \tfrac{1}{Z} \textstyle\prod_{k=1}^{K} \Psi_k(\mathbf{f}_{(k)}) \\ &= \tfrac{1}{Z} \textstyle\prod_{k=1}^{K} \prod_{m=1}^{M} \phi(\mathbf{J}_m^T \mathbf{f}_{(k)}; \alpha_m). \end{aligned} \tag{4.27}$$

Considering the equivalence of the graphical model to the Gibbs distribution we can express the probability density of the image as

$$p(\mathbf{f}) = \frac{1}{Z}\exp\left[\sum_{k=1}^{K}\sum_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{f}_{(k)};\alpha_m)\right]. \tag{4.28}$$

These models have found wide spread use across low level vision problems[Schmidt *et al.* (2010) for a list] due to their generic nature, however their generative properties are rarely examined. It is even argued or questioned in some literature as to whether some of the common applications of these models accurately capture the statistics of natural images well [Schmidt *et al.* (2010)]. That being understood we continue on with this line of reasoning as the Markov random field formulation provides a nice theoretical context or setting in which to initially formulate our models.

4.4.1 Inference

Inference using this model is often carried out using a Bayesian framework. In the Bayesian setting the goal is to find the true image $\mathbf{u}$ given the observed image $\mathbf{f}$ by maximising the posterior probability

$$p(\mathbf{u}|\mathbf{f}) \propto p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \tag{4.29}$$

where $p(\mathbf{f}|\mathbf{u})$ is the likelihood model and $p(\mathbf{f})$ is an image prior distribution such as that defined by equation 4.28. The likelihood model is the model of the noise process corrupting the image. The goal is to maximise the posterior distribution $p(\mathbf{u}|\mathbf{f})$

$$p(\mathbf{u}|\mathbf{f}) \propto \prod_{i=1}^{N}\exp\left[-\frac{1}{2\sigma^2}(f_i-u_i)^2\right]\prod_{k=1}^{K}\prod_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{u}_k;\alpha_m). \tag{4.30}$$

Maximizing the posterior distribution is equivalent to minimizing the log likelihood

$$\log p(\mathbf{u}|\mathbf{f}) \propto \sum_{i=1}^{N} -\frac{1}{2\sigma^2}(f_i-u_i)^2 + \sum_{k=1}^{K}\sum_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{u}_k;\alpha_m). \tag{4.31}$$

In matrix form MAP estimation is carried out by finding the vector $\mathbf{u}$ that minimizes

$$E(\mathbf{u}|\mathbf{f}) = -\frac{1}{2}(\mathbf{f}-\mathbf{u})^T\Sigma^{-1}(\mathbf{f}-\mathbf{u}) + \sum_{k=1}^{K}\sum_{m=1}^{M}\phi(\mathbf{J}_m^T\mathbf{u}_k;\alpha_m). \tag{4.32}$$

MAP estimation may be intractable and as a result prior work on application of the Bayesian framework has focussed on the use of approximate methods using belief propagation [Yedidia *et al.* (2003),Felzenszwalb & Huttenlocher (2004)], graph cuts [Komolgorv & Zabih (2004)] and simple gradient based optimization techniques [Roth & Black (2009)].

4.4.2 Optimisation

Our proposed model is considerably simpler than the majority of the state-of-the-art application of Markov random field models to problems of image filtering and denoising such as those proposed in [Roth & Black (2009)]. We consider this simplified model as we are not seeking to capture the statistics of natural images but instead to give a sound theoretical context underpinning our edge-preserving smoothing algorithm. In fact we return to the humble Gaussian potential to model our prior information. Albeit our viewpoint on the optimisation and inference is unique. We choose the potential functions to be Gaussian potentials

$$\phi(\mathbf{J}_m^T\mathbf{u}_{(k)}) = \phi(u_i,u_j) = e^{\frac{-(u_i-u_j)^2}{2\sigma_2^2}} \tag{4.33}$$

and with $M=1$ in equation 4.28. We use a simple gradient based optimisation strategy for which we require the gradient of the energy $E(\mathbf{u}|\mathbf{f})$. By considering one node of the graph we may write

$$E(u_i|f_i) = -\frac{1}{2\sigma^2}(f_i-u_i)^2 + \sum_{u_i\sim u_j}\phi(u_i,u_j) \tag{4.34}$$

Now if we differentiate with respect to a node u_i

$$\frac{\partial E(u_i|f_i)}{\partial u_i} \tag{4.35}$$

we obtain

$$\frac{\partial}{\partial u_i}\left[-\frac{1}{2\sigma^2}(f_i-u_i)^2\right]=\frac{f_i-u_i}{\sigma^2} \tag{4.36}$$

and

$$\frac{\partial}{\partial u_i}\sum_{u_i\sim u_j}\phi(u_i,u_j)=-\frac{1}{\sigma_2^2}\sum_{u_i\sim u_j}(u_i-u_j)\phi(u_i,u_j) \tag{4.37}$$

where the term $\sum_{u_i\sim u_j}(u_i-u_j)\phi(u_i,u_j)$ is recognised from the literature as the unormalised graph Laplacian [Chung (1997)]. In our implementation we normalise this term to recover the random walk graph Laplacian

$$(\Delta_{rw})(u)=\frac{\sum_{u_i\sim u_j}(u_i-u_j)\phi(u_i,u_j)}{\sum_{u_i\sim u_j}\phi(u_i,u_j)}. \tag{4.38}$$

Then we have the expression in graph form

$$\frac{1}{\sigma_2^2}L\mathbf{u}+\frac{1}{\sigma^2}(\mathbf{f}-\mathbf{u})=0 \tag{4.39}$$

where L is the normalized graph Laplacian, $L=I-M_{rw}$, and M_{rw} is the Nadaraya-Watson kernel estimator. To perform the denoising we formulate the problem as that of searching for the stationary point of the equation

$$L\mathbf{u}+\mu(\mathbf{f}-\mathbf{u})=0 \tag{4.40}$$

where $\mu=\frac{\sigma_2^2}{\sigma^2}$. We can rewrite this expression as

$$(I-M_{rw})\mathbf{u}+\mu(\mathbf{u}-\mathbf{f})=0 \tag{4.41}$$

and rearranging we have

$$\mathbf{u}-\frac{1}{1+\mu}M_{rw}\mathbf{u}-\frac{\mu}{1+\mu}\mathbf{f}=0. \tag{4.42}$$

As noted in [Zhou *et al.* (2005)] if we choose

$$\Delta t=\frac{1}{1+\mu}$$

then we have the optimiser given by

$$\mathbf{u}=(1-\Delta t)(I-\Delta t M_{rw})^{-1}\mathbf{f}. \tag{4.43}$$

4.4.3 The Weighting Matrix

We desire that the weights used to model the topological structure of the image capture the photometric similarities. The construction of the graph is motivated by the considerations that pixels within a neighbourhood are likely to have been generated by the same process and is independent of the absolute position within the image. Secondly the density of the pixels conditioned on a small neighborhood is assumed to be independent from the rest of the image; that is a Markov property. Typically Gaussian potentials have been used to model the process generating the densities in a local neighbourhood [Buades *et al.* (2005a); Coifman & Lafon (2006); Duchenne *et al.* (2008); Shi & Malik (2000)]. We form the weight matrix W with the entries given by the following

$$W_{ij} = \phi(u_i, u_j) \tag{4.44}$$

$$\tag{4.45}$$

We note that the Frobenius norm scaled by a half is the energy for the prior in the Bayesian model, that is

$$p(\mathbf{u}) \propto \exp\left(-\sum_{k=1}^{K} \phi(\mathbf{J}^T \mathbf{u}_k)\right) = \exp\left(-\frac{1}{2}\|W\|_F^2\right). \tag{4.46}$$

The resulting matrix W is symmetric and positive definite. The degree matrix can be defined as

$$D = \begin{cases} D_{ij} = d_i = \sum_j W_{ij} & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.47}$$

In graph theory normalisation of the weight matrix can be carried out in a number of ways [Chung (1997)]. We recall the normalised weight matrix M_n is the matrix whose entries are determined by

$$M_{n,ij} = \frac{W_{ij}}{\sqrt{d_i}\sqrt{d_j}} \tag{4.48}$$

and for the random walker kernel matrix M_{RW}

$$M_{rw,ij} = \frac{W_{ij}}{d_i}. \tag{4.49}$$

From the kernel matrices various graph Laplacian operators, can be defined as in table 4.1. In image processing the normalised Laplacian has been applied to problems in image segmentation [Shi & Malik (2000)] and image denoising [Zhang & Hancock (2006)]. The matrix M_{rw} defines a random walk on a graph [Meila & J (2001)]. It exhibits the Markov property [Nadler *et al.* (2005)]. We can study this matrix in order to derive algorithms for diffusion models on images. We note that this matrix contains the geometric information contained in the signal $\mathbf{f}$. The weightings in this kernel matrix directly model the local geometry defined by the immediate neighbours of each node. Indeed M_{rw} defines the Markov matrix where its entries are transition probabilities. Action of the matrix M_{rw}, which is the Nadaraya-Watson kernel estimator, on the image data results in smoothing of the image. This is the nonlocal means image denoising algorithm.

Laplacian	Matrix
Unormalised Laplacian	$L_{un} = D - W$
Normalised Laplacian	$L_n = I - D^{\frac{-1}{2}} W D^{\frac{-1}{2}} = I - M_n$
Random Walker	$L_{rw} = I - D^{-1}W = I - M_{rw}$

Table 4.1: Definition of Graph Laplacian Operators

4.4.4 PageRank Denoising

Consider the dynamics

$$\mathbf{u}^n = [cU + (1-c)M_{rw}]^n \mathbf{u}^0 \tag{4.50}$$

where $A = [cU + (1-c)M_{rw}]$ is known as the PageRank matrix. U is a uniform $N \times N$matrix with entries given by

$$U = \begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix} \tag{4.51}$$

and $U = \mathbf{l}\mathbf{e}^T$ where $\mathbf{e}$ is the uniform $N \times 1$ vector with entries given by $\mathbf{e} = [\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}]^T$ and with a slight abuse of notation $\mathbf{l}$, in this case, is the uniform $N \times 1$ vector with entries given by $\mathbf{l} = [1, 1, \ldots, 1]^T$. Applying this iteration to our data produces a weighted combination of a uniform signal $c\mathbf{e}$ and the local variation or detail in the image $(1-c)M_{rw}\mathbf{u}^n$. The parameter c is the probability of mixing the image intensities at each iteration. Varying this parameter produces different quality in the denoising of the image. In the theory of web page ranking the matrix M_{rw} models and captures the link structure of the world wide web. One then imagines a random web surfer who at each time step is at a web page deciding to follow a web page at the next time step according to the decision: with probability c they rest by jumping to a web page uniformly and at random with probability $(1-c)$ jumps to one of the hyperlinks on the web page [Ng *et al.* (01)]. In this setting c is termed the teleportation probability and chosen within the range 0.1 to 0.2 with a typical choice of 0.15. Convergence of the algorithm results in a stationary distribution which corresponds to the ranking of the web pages. As stated, in our image denoising problem the Markov matrix, M_{rw}, is the Nadaraya-Watson kernel estimator capturing the self-similarities in the image and c is a probabilistic mixing parameter.

4.4.5 Relation to Kernel Density Estimation

At the heart of the PageRank denoising algorithm is the per-pixel search for the stationary point of the equation

$$u_i - \frac{\sum_{u_i \sim u_j} \phi(u_i, u_j) u_j}{\sum_{u_i \sim u_j} \phi(u_i, u_j)} + \mu(f_i - u_i) = 0. \tag{4.52}$$

and

$$m(u_i) = u_i - \frac{\sum_{u_i \sim u_j} \phi(u_i, u_j) u_j}{\sum_{u_i \sim u_j} \phi(u_i, u_j)}. \tag{4.53}$$

The difference $m(u_i)$ is called the mean-shift in [Fukunaga & Hostetler (1975)] because it is the difference between the weighted mean of the neighbours u_j around u_i and the current value of u_i. This mean-shifting term was applied to nonparametric density gradient estimation. In the theory of nonparametric estimation it

is hypothesised that the gradient of a probability density function can be estimated using sample observations within a small region. In general nonparametric density estimation methods seek to model the process that generates the observed data by making few assumptions about the form of the underlying distribution. Kernel density estimation (KDE) is one of the more popular techniques in the pattern recognition literature [Bishop (2007)]. The utility and motivation for such methods comes from the fact that often in pattern recognition problems very little information, if any, is available about the nature of the underlying distribution from which the data is observed. Due to this lack of information nonparametric techniques rely on density estimated in localised regions.
Given N data points u_i, $i = 1 \dots N$, in a D-dimensional space and a kernel function

$$K = \phi(u_i, u_j) \tag{4.54}$$

then we can define a kernel density estimate as

$$p(\mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} \phi(u_i, u_{j.}) \tag{4.55}$$

We assume the kernel function to be the Gaussian potential. Differentiation with respect to u_i and normalisation leads to the density estimator to be given by the mean-shift $m(u_i)$ [Fukunaga & Hostetler (1975)]. The mean-shifting term has been applied in computer vision and pattern recognition [Fukunaga & Hostetler (1975),Cheng (1995),Comaniciu & Meer (2002),Carreira-Perpinan (2006),Paris & Durand (2008)] to empirically find modes of $p(\mathbf{u})$ in an iterative strategy without explicitly computing the distribution $p(\mathbf{u})$. Modes of the distribution correspond to peaks in the high dimensional data. Our algorithm can be interpreted as a combination of a mean-shift term and a localised difference term.

4.5 Implementation

As the matrix A, equation Markov matrix there is strong theory to suggest the existence and convergence to the stationary distribution [Haveliwala & Kamvar

(2003)]. By simple algebra the PageRank dynamics of equation 4.50 can be rewritten as

$$\mathbf{u}^{n+1} = c\mathbf{e} + (1-c)M_{rw}\mathbf{u}^n. \tag{4.56}$$

We prefer this formulation as we do not explicitly form and store the dense matrix U. Secondly we replace $\mathbf{e}$ with $\mathbf{f}$

$$\mathbf{u}^{n+1} = c\mathbf{f} + (1-c)M_{rw}\mathbf{u}^n. \tag{4.57}$$

This corresponds to what is known as personalised PageRank [Jeh & Widom (2003)]. From Markov theory if a stationary distribution exists then

$$\mathbf{u}^* = c\mathbf{f} + (1-c)M_{rw}\mathbf{u}^*. \tag{4.58}$$

where $\mathbf{u}^*$ is satisfied. Rearranging this expression we have

$$(I + (c-1)M)\mathbf{u}^* = c\mathbf{f}$$

and

$$(I - \Delta t M)\mathbf{u}^* = (1-\Delta t)\mathbf{f}$$

$$\mathbf{u}^* = (1-\Delta t)(I - \Delta t M_{rw})^{-1}\mathbf{f}. \tag{4.59}$$

Which is equation 4.43. We therefore have two algorithms for performing the denoising.

Algorithm 4 PageRank Denoising (Power Method)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W$
4: Choose $\Delta t,\ \alpha = 1 - \Delta t,\ \epsilon$
5: Initialise $\mathbf{f} \leftarrow \frac{\mathbf{f}}{\|\mathbf{f}\|_1},\ \mathbf{u} \leftarrow \mathbf{f}$
6: **for** $n = 0, 1, 2, \ldots \mathbf{u}^{n+1} - \mathbf{u}^n < \epsilon$ **do**
7: $\quad \mathbf{u} \leftarrow \alpha\mathbf{f} + (1 - \alpha)M\mathbf{u}$
8: **end for**

Algorithm 5 PageRank Denoising (Conjugate Gradient)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W$
4: Choose Δt
5: Solve $\mathbf{u} \rightarrow (1 - \Delta t)(I - \Delta t M_{rw})^{-1}\mathbf{f}$

4.6 Discussion

We have proposed a Markov random field formulation for graph-based regularisation and diffusion. This has the advantage of linking such strategy with other Markov random field formulations for image denoising. Moreover we related the algorithm to nonparametric statistics. Further to this our proposed PageRank denoising algorithm can be seen to be an extension of the non-local means algorithm proposed in [Buades *et al.* (2005a)]. Specifically it is a stable iterative non-local means algorithm. If we again consider the equation

$$\mathbf{u}^{n+1} = c\mathbf{f} + (1 - c)M_{rw}\mathbf{u}^n \tag{4.60}$$

we see that the non-local means is driving the diffusion process. In fact one might have been tempted to motivate the following iteration

$$\mathbf{u}^{n+1} = \Delta t M_{rw}\mathbf{u}^n \tag{4.61}$$

as an iterative non-local means strategy. Unfortunately such an algorithm suffers from two deficiencies

- choice of a suitable scale/time parameter Δt is not obvious
- an optimal stopping time for the algorithm is difficult to specify.

Figures 4.1, 4.2 and 4.3 illustrate the application of this iterative non-local means algorithm for different scale/time parameters. We notice that the algorithm diffuses the intensities in the image. Furthermore greater choices of the time/scale parameter increases the "speed" of the diffusion.

We notice that the powers of M_{rw} allow us to diffuse the image data:

$$\mathbf{u} \leftarrow \Delta t M_{rw}^{n} \mathbf{f}. \tag{4.62}$$

A classical approach to view the action of the powers of Markov matrix operators on a vector is through spectral analysis. We refer the reader to [Trefethen & Embree (2005)] for an introduction to this subject. From this theory of Markov chains as $k \to \infty$ we obtain a stationary distribution

$$\mathbf{p} = M^{\infty} \mathbf{u}^{0} \tag{4.63}$$

where

$$\mathbf{p} = M\mathbf{p}. \tag{4.64}$$

It is not difficult to show that the stationary distribution is given by

$$p_i = \frac{D_{ii}}{\sum_j D_{jj}} \tag{4.65}$$

which is the transpose of the left eigenvector of M_{RW}. Graph based clustering and denoising algorithms in signal processing are based upon solving the eigenvalue problem [Shi & Malik (2000), Meila & J (2001), Zhang & Hancock (2006)].

$$L\mathbf{x} = \lambda D\mathbf{x}. \tag{4.66}$$

Significance and relationship to semi-supervised learning. This approach reveals that we can diffuse image information along the graph. This behaviour has been used in the graph-based learning community to propagate class label information [Zhu *et al.* (2003a)] and for ranking hypertextual world wide web

Figure 4.1: Successive iterations (every 3^{rd}) of the update of equation 4.62 on the cameraman image. Top left is the original image, $\Delta t = 0.1$.

Figure 4.2: Successive iterations (every 3^{rd}) of the update of equation 4.62 on the cameraman image. Top left is the original image, $\Delta t = 0.5$.

Figure 4.3: Successive iterations (every 3^{rd}) of the update of equation 4.62 on the cameraman image. Top left is the original image, $\Delta t = 0.9$.

documents [Brin & Page (1998),Page *et al.* (1999)] but has not been taken full advantage of in the image processing literature. We recall that in the semi-supervised learning literature the problem is formulated as follows: given a set of data points

$$\{x_1, x_2, \ldots x_n\} \tag{4.67}$$

with a set of associated output labels

$$\{y_1, y_2, \ldots y_l\} \tag{4.68}$$

where $l < n$ and we are tasked with determining the labels of the remaining unlabeled data points

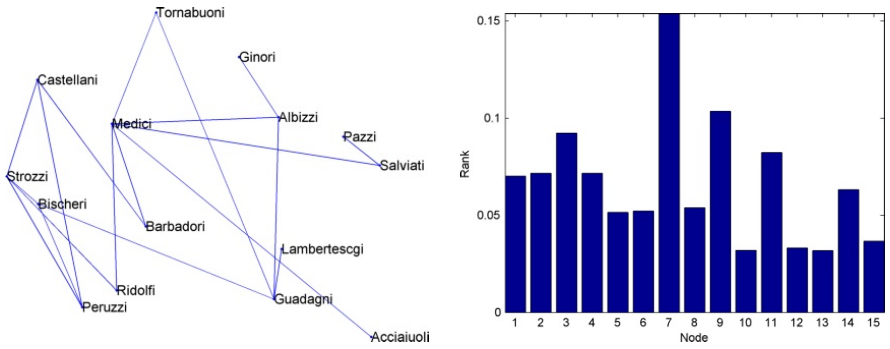
$$\{y_{l+1}, \ldots y_n\}. \tag{4.69}$$

A-priori we may have some information that allows us to generate weights that model the links between data points:

$$w_{ij} = \exp\left[-\frac{(x_i - x_j)^2}{\sigma^2}\right]. \tag{4.70}$$

Labeling the initial data is the semi-supervised step. However the labeled data is usually of a small or limited quantity as it may be a time consuming process to produce the labels whilst the unlabeled data may be abundant. Such problems are of practical importance and have been used to model link-spam detection and web-age ranking, email spam filtering [Zhu & Ghahramani (2002),Zhou *et al.* (2003),Zhou *et al.* (2004b)], video suggestion and discovery [Baluja *et al.* (2008)]. As an example of ranking figure 4.4 shows a graph of intermarriage relations between prominent 15^{th} century families. The data is obtained from [Newman (2005)]. Table 4.2 and figure 4.4 show the ranking of these families using the Google PageRank algorithm.

Underlying assumptions for these algorithms is that the data points are embedded on an unknown manifold. A weighted graph $G = \{\mathbb{V}, \mathbb{E}\}$ is used to represent the data points where the data points are the nodes of the graph. The similarities between the nodes are encoded in a weight matrix W. Given the graph G a simple idea is to propagate or diffuse the labeled nodes to unlabeled regions. For the



(a) Intermarriage relationships (b) PageRank score

Figure 4.4: The ranking of influence of 15^{th} century Florentine families using Google's PageRank score. (a) A graph of marriage links between families. (b) Bar graph of PageRank scores. 4.2 Table serves as a legend.

binary case we label the classes $\{+1, -1\}$. Many authors have proposed various label propagation algorithms [Zhu & Ghahramani (2002),Zhou *et al.* (2003),Zhou *et al.* (2004b)]. For illustration algorithm (6) reproduces the method proposed in [Zhu & Ghahramani (2002)]. We can observe that the mechanism for propagating

Algorithm 6 Label propagation (Zhu and Ghahramani)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W$
4: Initialise $\mathbf{y} \leftarrow (y_1, \ldots, y_l, 0, \ldots, 0)$
5: **for** $n = 0, 1, 2, \ldots$ until convergence **do**
6: $\quad \mathbf{y}^{n+1} \leftarrow M_{rw}\mathbf{y}^n$
7: $\quad \mathbf{y}_l^{n+1} \leftarrow \mathbf{y}_l^n$
8: **end for**

the labels is the Nadaraya-Watson kernel estimator, which we discussed in the context of nonparametric statistics. Lastly we would like to mention that in his framework we have propagated image intensities through the graph; in later chapter we will propagate image features in general.

Family	Node	PageRank Score
Medici	7	0.1538
Guadagni	9	0.1035
Strozzi	3	0.0923
Albizzi	11	0.0823
Castellani	4	0.0716
Bischeri	2	0.0716
Peruzzi	1	0.0702
Salviati	14	0.0632
Tornabuoni	8	0.0539
Barbadori	6	0.0522
Ridolfi	5	0.0515
Pazzi	15	0.0368
Ginori	12	0.0332
Lambertescgi	10	0.0321
Acciaiuoli	13	0.0319

Table 4.2: The ranking of influence of 15^{th} century Florentine families using Google's PageRank score. This list is in decreasing level of influence.

Chapter 5

Evaluations and Comparisons

5.1 Chapter Summary and Contributions

The objective of this chapter is to carry out a comprehensive evaluation of the PageRank denoising algorithm. We propose that the PageRank denoising algorithm is a solution of the Poisson equation on a graph. To facilitate the evaluation we develop another novel edge-preserving smoothing algorithm. This algorithm, which we call Power Iteration denoising (PID), is based upon the iterative application of the normalised pair-wise similarity matrix generated from the image data. In essence it is a stable application of a power iteration of the Nadaraya-Watson kernel estimator. In some sense it can be considered an iterative non-local means algorithm. More significantly it is an approximate solution of the Laplace equation on a graph.

The evaluation includes an analytic comparison between these methods and the iterative methods for graph regularisation. Further evaluation of these novel edge-preserving smoothing algorithms is carried out by application to some computational photography tasks. Many recent computational photography tasks require the decomposition of an image into a smooth base layer containing large scale intensity variations and a residual layer capturing fine details. Edge-preserving smoothing is the main computational mechanism in producing these multi-scale image representations. We, in effect, introduce a new approach to edge-preserving multi-scale image decompositions. Where as prior de facto approaches such as the Bilateral filter and weighted-least squares methods require multiple parameters

to tune the response of the filters our method only requires one. This parameter can be interpreted as a scale parameter. Furthermore our approach is theoretically related to diffusion processes on graphs. We demonstrate the utility of our approach by applying the method to computational photography tasks that utilise multi-scale image decompositions.
Finally one should note that all of the images used in the visual experiments are better appreciated when viewed full size on a computer monitor.

5.2 Introduction

We have seen in the previous chapter that graph based methods and non-local algorithms were introduced into the image processing literature from a differential geometry viewpoint. We then gave a probabilistic interpretation to these models and used a Bayesian formulation

$$p(\mathbf{u}|\mathbf{f}) \propto p(\mathbf{f}|\mathbf{u})p(\mathbf{u}) \tag{5.1}$$

with the log-likelihood function modeled as a multidimensional Gaussian distribution and the prior $p(\mathbf{u})$ as a Markov random field prior the derivation of the PageRank denoising algorithm. In deriving the denoising algorithm from this viewpoint we were considering the pixels to be corrupted by a zero mean white Gaussian noise process. It assumes that we have a likelihood model of the process generating the image noise, that is we made a Gaussian assumption. This assumption allowed us to derive the PageRank denoising algorithm and give a nonparametric statistical interpretation of the model. Here we formulate the algorithm using a conditional random field which is a discriminative model. The conditional random field model helps us to capture the intuition of our thinking. Indeed, in deriving the PageRank denoising algorithm we seek only to capture and model the self-similarities in neighbouring pixels. In the Bayesian formulation we explicitly assumed a Gaussian likelihood. This assumption allowed us to understand the PageRank algorithm as a generative model. Working only with the hypothesis that natural images exhibit strong dependencies in localised neighbourhoods we seek a model that captures, in a principled manner, the spatial interactions within a neighbourhood.

Given a graph $G = \{\mathbb{V}, \mathbb{E}\}$, the pixels of a noisy image $\mathbf{f}$ and the recovered 'clean' image $\mathbf{u}$ are indexed by the vertices of G. The construction enjoys the Markov property

$$p(u_i|\mathbf{f}_{V\setminus\{i\}}) = p(u_i|\mathbf{f}_{\mathcal{N}(i)}). \tag{5.2}$$

Therefore a conditional random field [Lafferty (2001),Kumar & Martial (2003),Kumar & Hebert (2006)] globally conditioned on the observed image $\mathbf{f}$ can be constructed, given by the distribution

$$p(\mathbf{u}|\mathbf{f}) = \frac{1}{Z}\exp\left(\sum_{i=1}^{N} A_i(u_i, f_i) + \sum_{k=1}^{K} \Psi_k(\mathbf{u}_{\{k\}})\right) \tag{5.3}$$

where A_i and Ψ_k are known as the association and interaction potentials respectively. The association potential models how likely a node u_i accepts a value f_i given the image $\mathbf{f}$ and ignoring the effects or interactions with the other nodes. The interaction potentil Ψ_k models how similar nodes in a neighbourhood interact.

If we choose the association potential to be a quadratic penalty term

$$A_i = \frac{\mu}{2}(f_i - u_i)^2 \tag{5.4}$$

and the interaction potential to be a Gaussian function

$$\Psi_{k=\{i,j\}} = \exp\left(-\frac{(u_i - u_j)^2}{\sigma^2}\right) \tag{5.5}$$

where $k = \{i, j\}$ refers to a clique with pixels indexed by i and j. When these measures are convex and positive preserving we recover the efficient optimisation strategy of the PageRank denoising algorithm.

Optimisation of equation (5.3) is performed by maximising the likelihood or minimising the log-likelihood of this distribution. Taking the log-likelihood of equation (5.3) we obtain

$$\log p(\mathbf{u}|\mathbf{f}) = \sum_{i=1}^{N} \frac{\mu}{2}(f_i - u_i)^2 + \sum_{k=1}^{K} \exp\left(-\frac{(u_i - u_j)^2}{\sigma^2}\right) - \log Z. \tag{5.6}$$

Then considering one node of the graph, u_i, and all pairwise cliques k for which $u_i \sim u_j$ then we can write the expression

$$\log p(u_i|f_i) = \frac{\mu}{2}(f_i - u_i)^2 + \sum_{u_i \sim u_j} \exp\left(\frac{-(u_i - u_j)^2}{\sigma^2}\right) - \log Z_i \tag{5.7}$$

and setting the derivative

$$\frac{d\left[\log p(u_i|f_i)\right]}{du_i} = 0 \tag{5.8}$$

we obtain

$$\mu(f_i - u_i) - \frac{1}{\sigma^2}\sum_{i \sim j}(u_i - u_j)\exp\frac{-(u_i - u_j)^2}{\sigma^2} = 0 \tag{5.9}$$

where we notice the right hand term to be a graph Laplacian and recalling the trick

$$\Delta t = \frac{1}{1 + \mu} \tag{5.10}$$

then we have derived the PageRank denoising algorithm as an optimiser for a conditional random field. What is interesting about this construction is that we have not imposed a Gaussian noise model a-priori and the construction sought only to establish self-similarities between the nodes. In essence the nodes could represent any type of image feature such as a pixel, an image derivative or even an image patch. All that is required is that one can define a similarity measure for the node data. Moreover this model captures the philosophy of the relationship of the PageRank denoising algorithm to kernel density estimation. Before we compare and evaluate the PageRank denoising algorihtm we develop a simple power iteration strategy to effect image denoising and edge-preserving smoothing.

5.3 Power Iteration Denoising

Essentially this algorithm finds a low dimensional data embedding using a truncated power iteration of a normalized pair-wise similarity matrix generated from the image. This algorithm is essentially a stable iterative non-local means [Buades *et al.* (2005a)] method. Our method is an application of the recently introduced *power iteration clustering* method [Cohen & Lin (2010)] for graph clustering to the image denoising problem.

5.3.1 Basic Algorithm

It is well established in the literature [Cohen & Lin (2010); Meila & J (2001)] that the matrix M_{rw} is closely related to the normalized random walk Laplacian $L_{\text{rw}} = I - D^{-1}W = I - M_{\text{rw}}$ The relationship between these matrices is noted in the eigenvalues and eigenvectors of these two matrices. [Meila & J (2001)] and [Shi & Malik (2000)] established that the second smallest eigenvector of L_{rw}, that is the eigenvector associated with the second smallest eigenvalue [1], defines a partition of the graph that approximately maximizes the Normalized cut criterion. Generally the k smallest eigenvectors defines a subspace where the clusters in the graph are well separated. As established in [Meila & J (2001)] the k smallest eigenvectors of L_{rw} are equivalent to the k largest eigenvectors of M_{rw}. To compute these k largest eigenvectors of M_{rw} one can run a *power iteration* (PI) [Cohen & Lin (2010)]. PI is an iterative update which starts with an arbitrary vector $\mathbf{u}^0 \neq 0$ and repeatedly performs the update

$$\mathbf{u}^{n+1} \leftarrow cM_{rw}\mathbf{u}^n \tag{5.11}$$

where c is a normalizing constant that keeps $\mathbf{u}^n$ from getting too large. The parameter c is chosen such that $c \leftarrow \frac{1}{\|M_{rw}\mathbf{u}^n\|_1}$
For the image denoising problem we choose $\mathbf{u}^n \leftarrow \|\mathbf{f}\|_1$ that is the original noisy image is normalized as a probability distribution. The main idea of the algorithm is that whilst running PI to convergence does not produce interesting results in that it equalizes the image intensities, it is in fact the intermediary vectors that are of interest. In essence we require an early termination of the algorithm to produce an effective denoising algorithm. One might run (PI) for a small number of iterations and locally embed or trap the image as an eigenvector of M_{rw}, that is well before final convergence.
For the spectral clustering problem [Cohen & Lin (2010)], it was suggested that this could be effected by defining the velocity at iteration n to be the vector $\delta^n = \mathbf{u}^n - \mathbf{n}^{n-1}$ and the acceleration at iteration n to be the vector $\epsilon = \delta^n - \delta^{n-1}$ and stop the PI when $\|\epsilon\| < \hat{\epsilon}$ where $\hat{\epsilon}$ is a small threshold. This heuristic is based on the assumption that while the clusters are "locally convergent" the rate

[1] This eigenvector is often referred to as the Fiedler vector.

of convergence changes rapidly where as during global convergence the convergence rate appears more stable. The complete denoising algorithm is presented in algorithm (7). In all our experiments we use σ equal to the noise standard

Algorithm 7 PID (power iteration denoising)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W,\ \mathbf{u}^0 \leftarrow \|\mathbf{f}\|_1$
4: **for** $n = 0, 1, 2, \ldots$ until $\|\delta^{n+1} - \delta^n\| < \hat{\epsilon}$ **do**
5: $\quad \mathbf{u}^{n+1} \leftarrow cM_{rw}\mathbf{u}^n$
6: $\quad \delta^{n+1} \leftarrow \mathbf{u}^{n+1} - \mathbf{u}^n$
7: **end for**
8: output: $\mathbf{u} \leftarrow \frac{\mathbf{u}}{\max(\mathbf{u})}$

deviation of the corrupted image and $\hat{\epsilon} = 1 \times 10^{-5}$.

5.3.2 A conditional random field formulation

For our image graph G we can associate a set of k fully connected pair-wise cliques (fully connected subgraphs). We can formulate the PID algorithm as the maximization of the conditional random field model

$$p(\mathbf{u}|\mathbf{f}) = \frac{1}{Z}\exp\left(-\sum_{k=1}^{K}\Psi_k(\mathbf{u})\right). \tag{5.12}$$

where Z is a normalization constant and Ψ_k is known as the interaction potential. We have defined this model over every pair-wise clique with potentials given by

$$\Psi_k = \frac{w_{i,j}}{2\sum_j w_{i,j}}(u_i - u_j)^2. \tag{5.13}$$

Inference with this model is carried out by maximizing the conditional distribution $p(\mathbf{u}|\mathbf{f})$ or equivalently minimizing the log-likelihood of the distribution. We therefore minimize the energy

$$E(\mathbf{u}|\mathbf{f}) = -\frac{1}{2}\mathbf{u}^T(I - M_{rw})\mathbf{u} \tag{5.14}$$

where $L_{rw} = I - M_{rw}$ is the random walk graph Laplacian. Using a fixed point iteration we minimize this function using the strategy

$$\mathbf{u}^{n+1} \leftarrow cM_{rw}\mathbf{u}^n \tag{5.15}$$

and choosing $c \leftarrow \frac{1}{\|M_{rw}\mathbf{u}\|_1}$ we recover the PID algorithm. This algorithm is a direct result of the probabilistic arguments we gave to the differential geometric models for the graph based models of [Elmoataz *et al.* (2008)]. The PID algorithm can be related to the PageRank denoising algorithm through the conditional random field formulation. The energy minimisation of the PID algorithm is a relaxation of the conditions imposed in the PageRank denoising algorithm. Specifically if we drop the association potential of equation 5.3 and impose weighted interaction potentials we can then recover the PID algorithm. Moreover the PID algorithm can be considered as a solution of the Laplace equation

$$L_{rw}\mathbf{u} = 0 \tag{5.16}$$

on a graph.

5.3.3 Comparisons with related methods

If we recall recent formalisms of graph methods for image denoising, and processing in general, their justification is as the result of a differential geometric argument. [Elmoataz *et al.* (2008)] proposed graph regularisation using the following iteration

$$u_i^{n+1} = \frac{\lambda f_i + \sum_j \phi(u_i, u_j) u_j^n}{\lambda + \sum_{u_i \sim u_j} \phi(u_i, u_j)}. \tag{5.17}$$

This expression can be written as

$$u_i^{n+1} = \frac{\lambda f_i}{\lambda + \sum_{u_i \sim u_j} \phi(u_i, u_j)} + \frac{\sum_j \phi(u_i, u_j) u_i^n}{\lambda + \sum_{u_i \sim u_j} \phi(u_i, u_j)} \tag{5.18}$$

and in matrix form

$$\mathbf{u}^{n+1} = \lambda(\lambda + D)^{-1}\mathbf{f} + (\lambda + D)^{-1} W \mathbf{u}^n. \tag{5.19}$$

The λ term is chosen as a heuristic to stabilise the iteration. We notice that for $\lambda = 0$ we have have an iterative application of the Nadaraya-Watson kernel that is an iterative application of the non-local means. This can recover the PID algorithm. [Elmoataz *et al.* (2008)] could not justify a suitable termination criterion for such an iterative strategy which we provided for the PID algorithm. For values of $\lambda > 0$ the iteration of equation 5.19 is a heuristic argument for the solution of the discrete Poisson equation on a graph

$$L\mathbf{u} = \mathbf{f} \tag{5.20}$$

using the Gauss-Jacobi iterative strategy. To see this we can use the operator split $L = D - W$ and write the Gauss-Jacobi iteration

$$D\mathbf{u}^{n+1} = W\mathbf{u}^n + \mathbf{f} \tag{5.21}$$

and finally

$$\mathbf{u}^{n+1} = D^{-1}W\mathbf{u}^n + D^{-1}\mathbf{f}. \tag{5.22}$$

[Elmoataz *et al.* (2008)] proposed the introduction of the parameter λ, giving the iteration of equation 5.19, in order to stabilise the iteration. This heuristic

argument introduces the difficulty of choosing the parameter λ. In their paper the advocate the choice of a "small" value; typical values in their experiments being $\lambda = 0.05$ and $\lambda = 0.005$. In our experiments we will show that our PageRank denoising algorithm can be considered a "close" approximation to the solution of the Laplace equation

$$L_{rw}\mathbf{u} = \mathbf{f} \tag{5.23}$$

and does not require the use of heuristic arguments.

5.4 Experiments

Before providing experimental results we consider the image data sets and performance measures used for experimentation and evaluation. We use three image datasets for the evaluation of the denoising algorithms. The first two datasets are standard image datasets from the image processing literature [Portilla *et al.* (2003),Martin *et al.* (2001)]. The third dataset is a collection of noisy digital camera images.

Figure 5.1 shows standard images for comparing image denoising algorithms [Portilla *et al.* (2003)]. From left to right and top to bottom the figures are known as Barbara, Boats, Cameraman, House, Lena and Peppers. For further experimental evaluation we use a second dataset comprising a subset of images, 16, taken from the Berkeley image dataset [Martin *et al.* (2001)] which provides corresponding colour and greyscale images. The subset of images that we use are shown in figures 5.2 and 5.3 containing various natural images of outdoor scenes, buildings, animals, people, indoor scenes, textures and combinations. This allows us to benchmark our algorithm on a variety of natural image types. We test our proposed methods on these image datasets with computer-generated additive Gaussian white noise at 20 different variances.

Our experiments are carried out in two phases: the first phase is to understand the relative contribution of various aspects of our methods. In particular we seek to investigate computational aspects of the variants of the PageRank denoising algorithms: algorithms 4 and 5 in chapter 4.5 and the sensitivity of the algorithm to the parameter c in the iteration

$$\mathbf{u}^{n+1} \leftarrow c\mathbf{f}^n + (1-c)M_{rw}\mathbf{u}^n. \tag{5.24}$$

Experiments 1 and 2 are described later in this chapter and pertain to the first phase of experimentation. The second phase of the experiments are designed to provide quantitative benchmarks and visual comparisons. These experiments use computer-generated synthesized image data and noisy digital camera images. Experiments 3 and 4 pertain to phase 2 of the experimentation.

Figure 5.1: Standard images for comparing image denoising algorithms. From left to right and top to bottom the figures are known as: Barbara, Boats, Cameraman, House, Lena and Peppers.

Figure 5.2: Subset of natural images take from the Berkeley segmentation dataset Martin *et al.* (2001).

24004 208001

285079 302003

Figure 5.3: Subset of natural images take from the Berkeley segmentation dataset Martin *et al.* (2001).

5.4.1 Quantitative Measures

To evaluate the performance of the denoising algorithm we need to rely on quantitative measures of the recovered images. We appeal to two measures

- PSNR (peak signal-to-noise ratio)
- SSIM (structured similarity index)

5.4.1.1 The PSNR measure

This measure is computed as

$$PSNR = 20\log_{10}\frac{1}{\sigma_e} \tag{5.25}$$

where σ_e is the standard deviation of the pixel-wise image error; that is

$$\sigma_e = \sqrt{\frac{1}{N-1}(\mathbf{f}-\mathbf{u})^T(\mathbf{f}-\mathbf{u})} \tag{5.26}$$

where $\mathbf{f}$ is the target image, $\mathbf{u}$ is the recovered image and N is the number of pixels. The image intensites are normalised to values in the range $[0, 1]$. The measure is evaluated in decibels, dB. A factor 2 reduction of noise is an approximate increase of 6dB in PSNR.

5.4.1.2 The SSIM Measure

Although the PSNR is a widely used metric to evaluate image quality of denoised images it has the draw back that it does not take into consideration the perceptual quality of an image to a human observer, that is we realise natural images are highly structured with pixel elements containing strong dependencies. As a result we require a score that takes these ideas into consideration. Here we outline the salient features of the SSIM index. Suppose we have a faithful reference image f and a corrupted version u then to compute the SSIM index we use the measuring:

$$S(f,u) = \frac{(2\mu_f\mu_u + C_1)(2\sigma_f\sigma_u + C_2)}{(\mu_f^2 + \mu_u^2 + C_2)} \tag{5.27}$$

where μ_f and μ_u are unbiased estimates of the mean image intensities of the signals f and u respectively, σ_f and σ_u are unbiased estimates of the standard

deviation of the image intensities of f and u. The parameter is a measure of the correlation of the two signals and C_1 and C_2 are constants. The measure is symmetric

$$S(f,u) = S(u,f) \tag{5.28}$$

and bounded

$$0 \leq S(f,u) \leq 1 \tag{5.29}$$

with a unique maximum. It is derived as a nonlinear combination of signal luminance, contrast and correlations. Intuitively the SSIM measure seeks to estimate the similarities between the reference and target image by quantitatively measuring:

1. the differences between the luminance or brightness values
2. the differences in the contrasts
3. the difference in the image structures

and combining these values in to a score between 0 and 1. A score of 1 indicating that the images are identical and a score of 0 being the extreme opposite. Experiments with human subjects demonstrated that this measure was perceptually more similar to the human visual system than traditional measure such as the PSNR. Practically the measure is computed in local 8×8 windows for which the SSIM is generated. A mean measure, the MSSIM, is then computed as an average (weighted average) of these windows. For full details of the parameters and constants we refer the reader to the paper[Wang *et al.* (2004)]. A MATLAB implementation is available at [Wang].

5.4.2 Phase 1

5.4.2.1 Experiment 1

The PageRank denoising algorithm can be solved in one of two ways. Firstly it can be solved as the stationary distribution of the Markov chain. This leads to the power method for solution of the linear system (algorithm 4). To make it explicit the power method is given by the stationary distribution of the linear system

$$\mathbf{u}^{n+1} \leftarrow c\mathbf{f} + (1-c)M_{rw}\mathbf{u}^n. \tag{5.30}$$

The convergence rate of the power method is given by the ratio $\frac{|\lambda_2|}{|\lambda_1|}$ [Golub & Van Loan (1996)] where λ_1 and λ_2 are the smallest and second smallest eigenvalues associated with this matrix system. It is known that $|\lambda_2| \leq (1-c)$ [Haveliwala & Kamvar (2003)] therefore rate of convergence is contingent on this parameter c. Secondly we can compute the solution of the linear system. The linear system approach (algorithm 5) requires the solution of the matrix system

$$(I - \Delta t M_{rw})\mathbf{u} = (1-\Delta t)\mathbf{f}. \tag{5.31}$$

The matrix system $(I - \Delta t M_{rw})$ can be very large, is sparse and symmetric. Solution of the linear system by a direct method is not feasible due to the matrix size and computational resources required. Direct methods introduce fill-in to the non-zero elements of the matrix system hence the memory requirements dramatically increase. We therefore concentrate on iterative methods, particularly the Krylov subspace methods. As this system is symmetric positive definite the ideal candidate, at first glance, is the conjugate gradient method (CG) [Golub & Van Loan (1996)]. Moreover the eigenvalues of the matrix $I - \Delta t M_{rw}$ are well clustered around 1 implying convergence is guaranteed [Chan & Jin (2007)]. We therefore choose to compare this method with its variants: conjugate gradient squared (CGS), biconjugate gradient (BiCG) and other Krylov subspace methods: minimum residual (MINRES) and quasi-minimum residual (QMR). For detailed discussions on these methods we refer the reader to [Barrett *et al.* (1994),Saad (2003), Benzi *et al.* (2005), Golub & Van Loan (1996),]. Table (5.1) shows the computational requirements of these algorithms.

Method	IP	SAXPY	SpMV	Storage
Power		1	1	Sparse Matrix + $2N$
CG	2	3	1	Sparse Matrix + $6N$
CGS	2	6	1	Sparse Matrix + $11N$
BiCG	2	5	1	Sparse Matrix + $10N$
MINRES	$n+1$	$n+1$	2	Sparse Matrix + $(i+5)N$
QMR	2	$8+4$	1	Sparse Matrix + $16N$

Table 5.1: Computational requirements. Operations per iteration i: IP (inner product) counts, SAXPY (sparse scalar alpha times vector X plus Y), SpMV (sparse matrix vector multiplication) per iterations, and storage counts the number of matrices and vectors required for the method. Some QMR implementations may require less SAXPY operations when the residual is not recursively updated. We use BLAS parlance [Barrett *et al.* (1994)].

For the experimental set up we use the set of images from the Berkeley image dataset in figures 5.2 and 5.3. We perform computational experiments on four different image sizes: 64×64, 128×128, 256×256 and 512×512 in order to understand the scalability of the algorithms. Firstly we take the images from the data set and crop a square region and scale that region to the appropriate size. We then take a solver, for example, the conjugate gradient method and a value for the mixing parameter Δt. We then run the PageRank denoising algorithm at the specified image size, for the specified solver and mixing parameter. We measure the number of iterations and time taken until convergence of the algorithms. The experiment is repeated for all images in the dataset and the measured outputs are averaged. The mixing parameter is then varied and the experimental process is repeated. The mixing parameter is varied in the range $[0.1 - 0.99]$. Convergence of the algorithm is chosen to be the residual $\|\mathbf{u}^{n+1} - \mathbf{u}^n\| \leq 1 \times 10^{-5}$ for the power method and $\|(I - \Delta t M_{rw}\mathbf{u}^n) - (1 - \Delta t)\mathbf{f}\| \leq \times 10^{-5}$ for the linear solvers. We set the maximum allowable number of iterations for each algorithm to be 200. If the algorithm has not converged within 200 iterations it is terminated. This process is repeated for all image sizes. The results are given in figures 5.4 and 5.4.2.1.

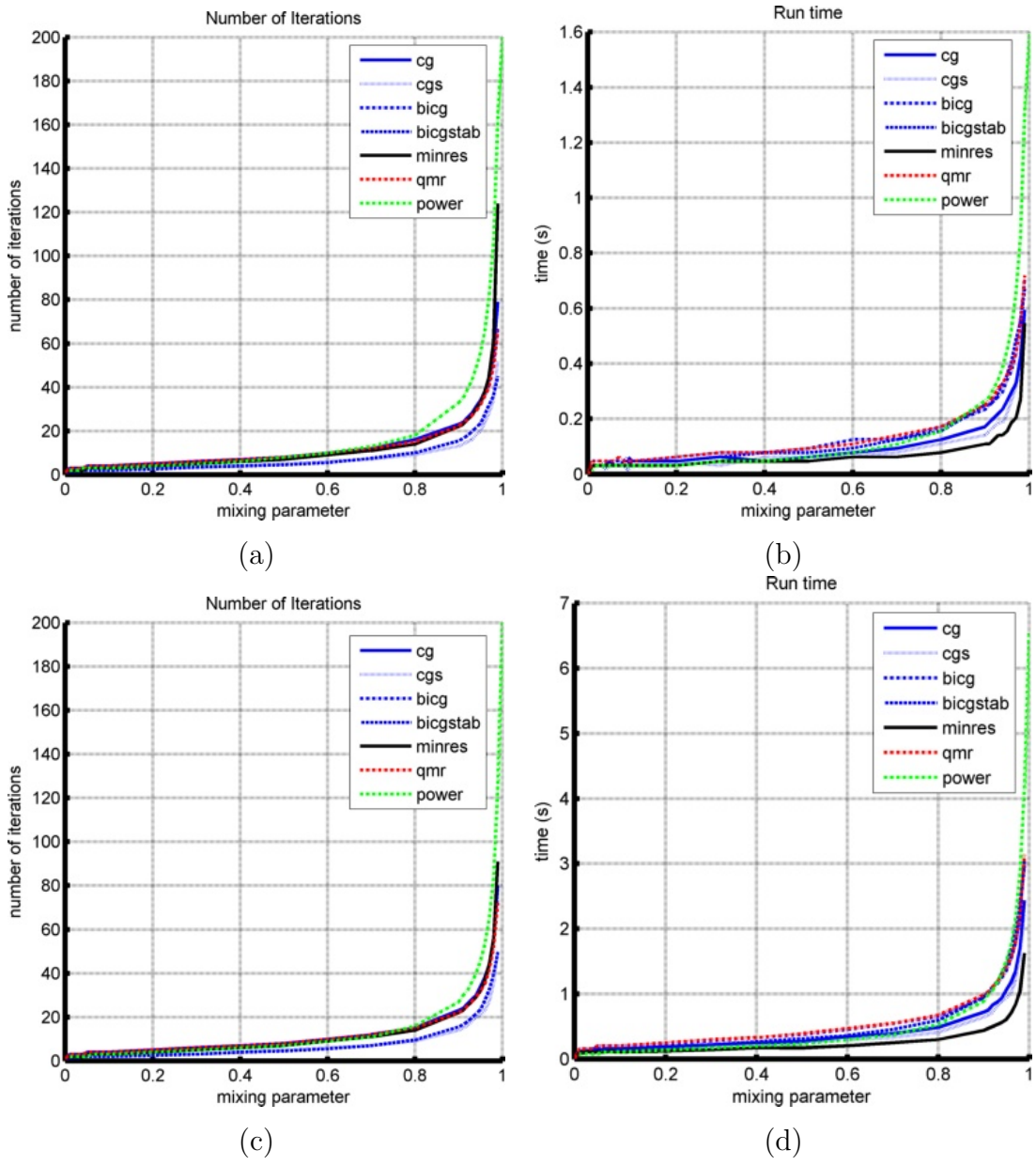


Figure 5.4: Convergence of power and iterative methods. Figures (a) and (b) are the number of required iterations and run times for images of size 64×64. Figures (c) and (d) are the number of required iterations and run times for images of size 128×128.

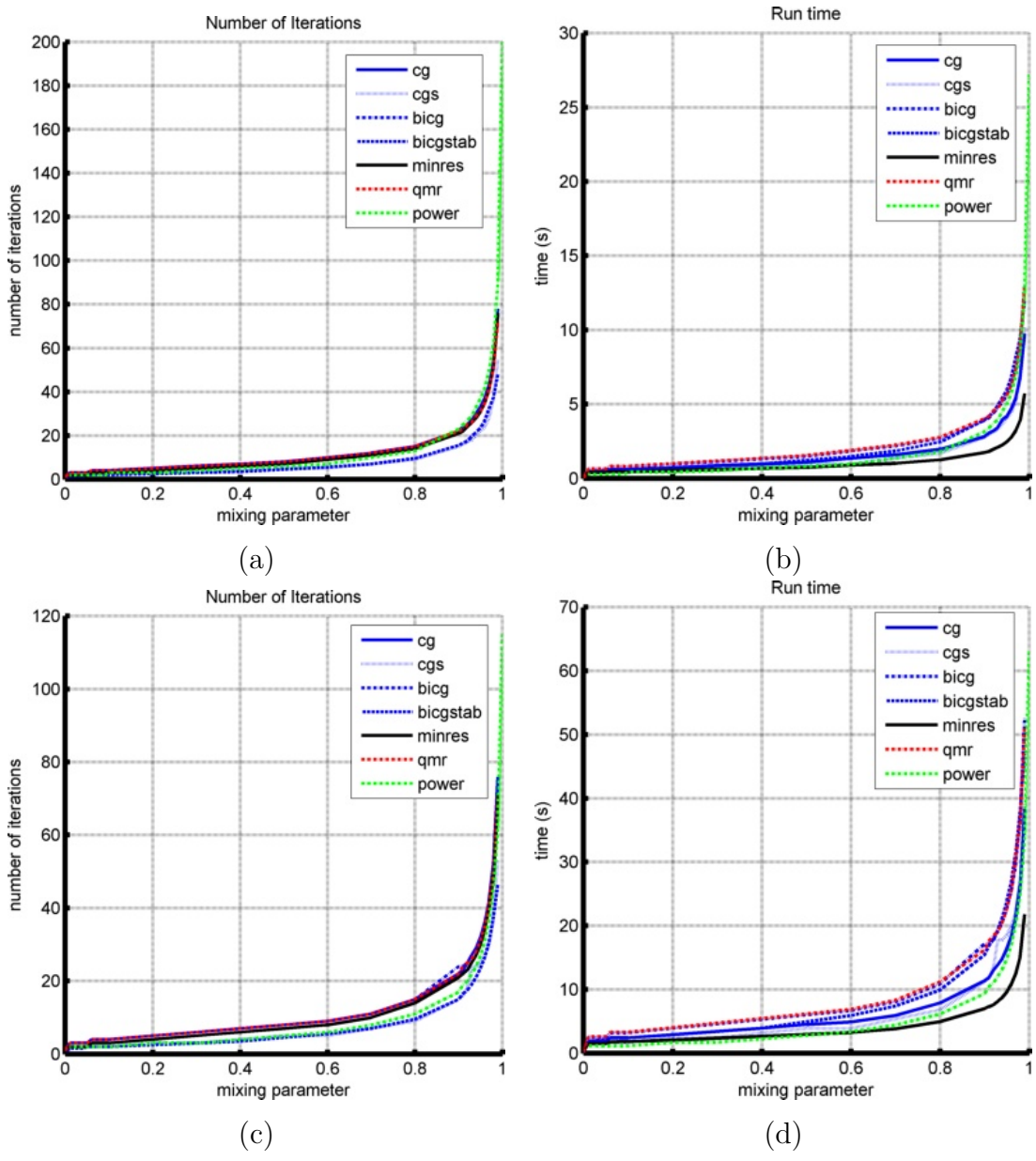


Figure 5.5: Convergence of power and iterative methods. Figures (a) and (b) are the number of required iterations and run times for images of size 256×256. Figures (c) and (d) are the number of required iterations and run times for images of size 512×512.

An analysis of the results reveals that:

(1) For all solvers and at all image sizes the number of iterations and algorithm run time increases asymptotically to ∞ as the mixing parameter $\Delta t \to 1$. This is consistent with the idea that as $\Delta t \to 1$ the matrix $(I - \Delta t M_{rw}) \to I - M_{rw} = L_{rw}$. As the matrix M_{rw} is symmetric the matrix M_{rw} has row sums equal to zero. Consequently the determinant of L_{rw} is 0 and the matrix is singular. In essence we are solving the Poisson equation on the graph.

(2) The power method does not always converge to the desired tolerance in less than 200 iterations. This is evident in the graphs of figures 5.4 (a), (c) and 5.4.2.1 (a).

(3) All the linear solvers converge in less than 100 iterations.

(4) In general the conjugate gradient method and its variants have similar performance.

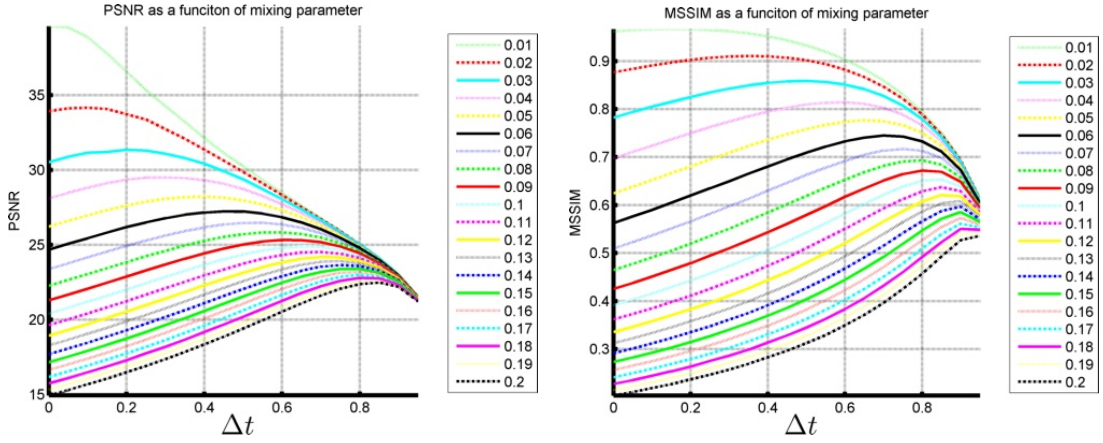
(5) The minimum residual and quasi-minimal residual methods are the best performing linear solvers. Minimum residual in particular being the fastest and most stable solver for all values of the mixing parameter and image sizes.

5.4.2.2 Experiment 2

For this second experiment we would like to understand the variation in the quality of the recovered image versus the mixing probability c in the PageRank denoising algorithm. We recall that

$$\begin{aligned} \mathbf{u}^{n+1} &\leftarrow c\mathbf{f} + (1-c)M_{rw}\mathbf{u}^n \\ c &= 1 - \Delta t. \end{aligned} \tag{5.32}$$

We abuse notation slightly as we refer to both c and Δt as the mixing parameter. We try and make explicit as to which we refer to when it may be ambiguous. For each image from the Berkeley segmentation dataset, figures 5.2 and 5.3, we



(a) (b)

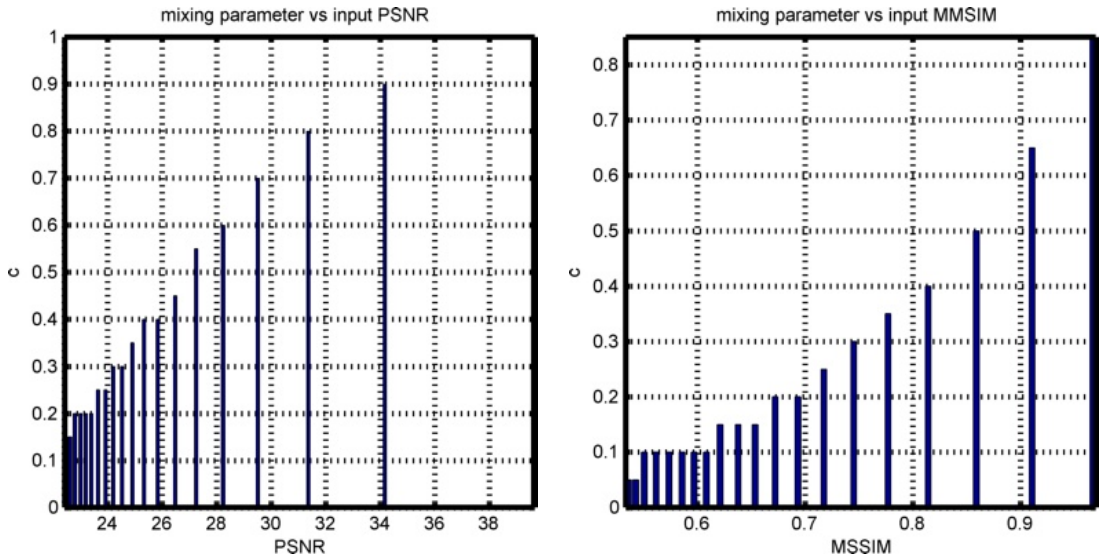
Figure 5.6: Output image quality as a function of the mixing parameter Δt. The legend shows the noise standard deviation corrupting the input images.

corrupt the image with zero mean Gaussian white noise. The noise is chosen to have variance in the range $[0.1 - 0.2]$. For each noise level we run the PageRank denoising algorithm for values of c in the range $[0 - 1[$ and observe the output PSNR and MSSIM measures. The results for each image are averaged over the entire dataset. The plots in figure 5.6 show the results of these experiments. The plots in figure 5.7 show the optimal c (the value of c for which PSNR and MSSIM are maximal) versus PSNR and MSSIM respectively. An analysis of the results show that:

(1) As $\Delta t \rightarrow 1$ the PSNR and MSSIM of the recovered images varies nonlinearly.

(2) There is a maximum peak performance or optimal value of the mixing parameter Δt for which the PSNR and MSSIM of the recovered image is maximized.

(3) As $\Delta t \rightarrow 1$ the PSNR and MSSIM of the recovered images tends to a similar value irrespective of the noise level corrupting the input image. It is in this region that we are solving the Poisson equation on the graph.



(a) (b)

Figure 5.7: Optimal mixing parameter as a function of PSNR and MSSIM of the recovered image.

(4) The plots of the optimal mixing parameter $c = 1 - \Delta t$ versus the recovered PSNR and MSSIM in figure 5.7 show that this parameter decreases with a decreasing quality of input image.

For the second part of the experiment we fix the value of the mixing parameter Δt. We choose a value of 0.9 and observe the quality of the recovered image for various value σ of the interaction potentials (equation 5.5) used to construct the graphs for the Power iteration denoising and PageRank denoising algorithms. Figure 5.8 shows the results of the experiments. We notice that:

(1) For each noise standard deviation there is a prescribed maximum σ for which there is a well defined maximum output PSNR/SSIM

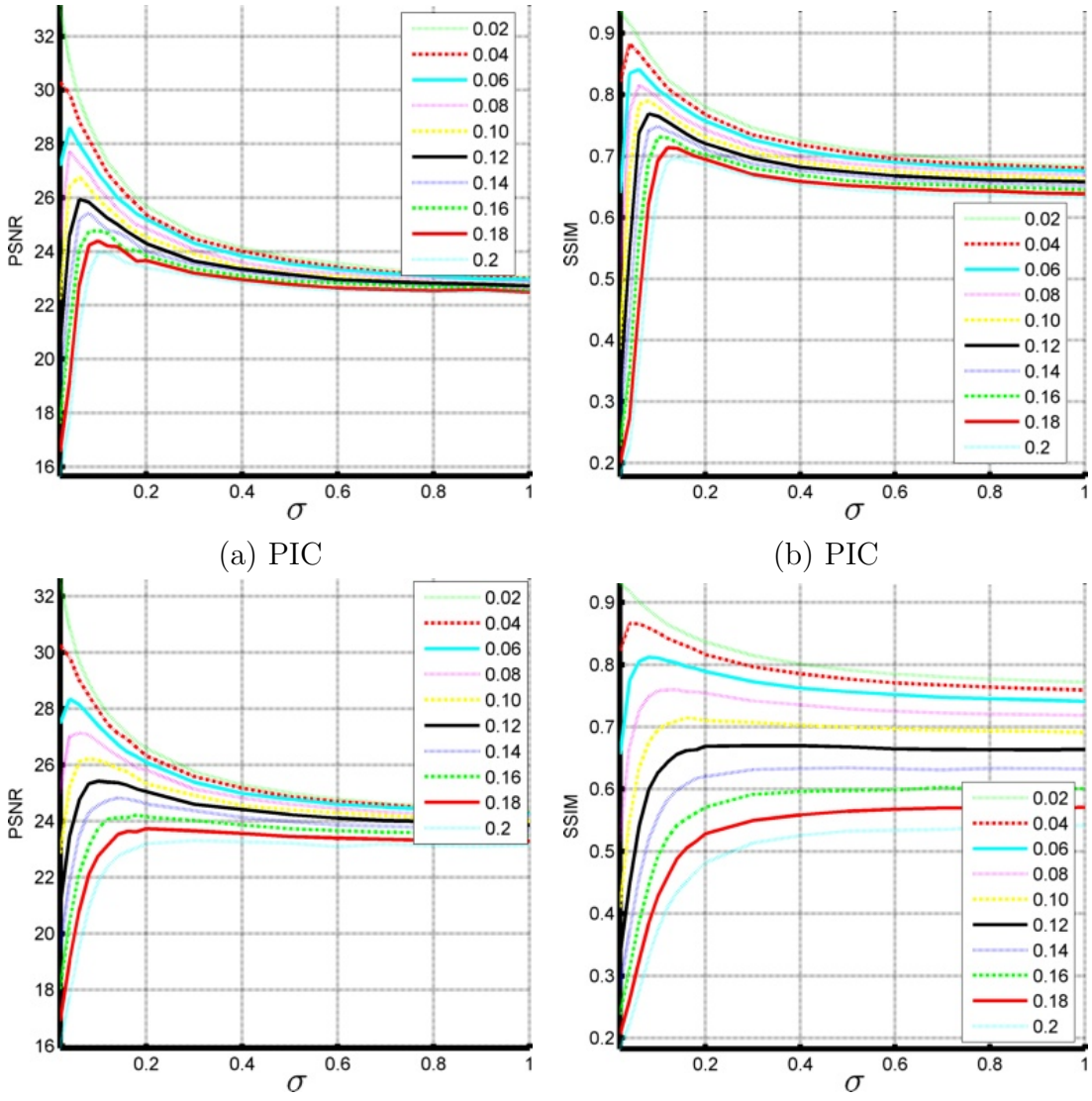
(2) As the size of σ increases after the defined maximum the performance of the filter tends to a constant value. This constant value can be attributed to the fact that as σ is increased the shape of the Gaussian potentials used to capture

the pair-wise pixel similarities has a less defined peak. In essence it tends to a flat shape. As a result the Laplacian tends to fail to capture the topological structure of the image and tends to an isotropic operator. In essence it becomes the standard linear Laplacian operator generated as the discretisation of the associated partial differential equation.

5.4.3 Phase 2

5.4.3.1 Experiment 3

To facilitate the comparisons we corrupt the images with zero mean additive Gaussian white noise. It is understood that this is not the noise process that typically affects modern digital camera systems but this is fairly standard practice in the literature for benchmarking denoising algorithms [Portilla *et al.* (2003),Buades *et al.* (2005a),Elmoataz *et al.* (2008),Schmidt *et al.* (2010)]. Recall that we are using the corresponding grey level images from the Berkeley segmentation database images of figures 5.2 and 5.3 with the image intensities normalised between 0 and 1. The noise standard deviation is varied in the range $[0.02 - 0.2]$ and the PSNR and MSSIM scores of the corrupted image (input PSNR) and the image output by the various denoising algorithms are measured. We average the respective input and output PSNR and MSSIM for the various noise levels over the entire dataset and report the average values. The results are plotted in figure 5.9 where figures 5.9 (a) and (b) are the recovered PSNR and MSSIM scores where the experiments are performed on the images of figure 5.1 and figures 5.9 (c) and (d) are the recovered PSNR and MSSIM scores where the experiments are performed on the images of figures 5.2 and 5.3. We chose to benchmark our algorithm against the Bilateral filter [Tomasi & Manduchi (1998)], non-local means [Buades *et al.* (2005a)] and total variation denoising algorithms [Chambolle (2004b)].
The Bilateral filter algorithm is the de facto denoising method in computer vision application areas such as computational photography. In particular it has been used as an edge-preserving filter for the removal of noise for images acquired by digital cameras [Petschnigg *et al.* (2004), Paris & Durand (2009)]. Secondly we choose the non-local means algorithm as mechanistically it is the most closely related denoising algorithm to our methods PID and PR. Specifically it can be



(a) PIC (b) PIC

(c)PR $\Delta t = 0.9$ (d) PR $\Delta t = 0.9$

Figure 5.8: Performance of algorithms PIC and PR versus varying σ. (a) and (b) Recovered PSNR and MSSIM versus σ for PIC algorithm. (c) and (d) Recovered PSNR and MSSIM for PR algorithm.

seen to be a special case of the PID algorithm. For a faithful comparison with this method we limit to the pixel-wise case. It has been noted that the results are improved if comparisons in the non-local means mechanism are effected at the patch level [Buades *et al.* (2005a)]. That being said our thesis is motivated by algorithms that operate exclusively on the pixel level. We use the parameter selection as suggested in [Buades *et al.* (2005a)] and limit the algorithm to pairwise cliques in 5×5 neighbourhoods. Lastly we have chosen the total variation as a becnhmark as it is one of the state-of-the art partial differential equation based diffusion algorithms.
In the PID algorithm we set the parameter σ = noise standard deviation. For the PR denoising algorithm we choose $\sigma = 1 -$ noise standard deviation and the mixing parameter c is set to the value that maximises the output PSNR and MSSIM scores according to the curves of figure 5.7. From the results we observe that:

(1) The non-local means, PID and PR denoising algorithms have a more linear response curves than the total variation and Bilateral filtering algorithms.

(2) The PID algorithm is the best performing algorithm on all datasets and performance measures.

(3) In the extensive evaluation over the Berkely image datasets, figures 5.9 (c) and (d) the PR denoising algorithm outperforms the non-local means algorithm in terms of the PSNR measure but is outperformed by the non-local means algorithm when the perceptual SSIM measure is used.

Finally we make note of the fact that when the images are corrupted with Gaussian noise with a low standard deviation (high input PSNR or SSIM) the output image may have a lower PSNR or SSIM score than the input. This effect is common to edge-preserving smoothers. As the noise level is low the dominant effect of the edge-preserving smoothing is to produce "cartoon-like" effects or image abstractions. Such characteristics and abstractions are desirable in some applications in computational photography such as non-photorealistic rendering and

image abstraction [DeCarlo & Santella (2002),Winnemller *et al.* (2006)].
An interesting direction for further study would be to develop a benchmark dataset that accurately models the noise processes that corrupt modern digital cameras. One could produce experiments that investigate the relationship between input PSNR (SSIM) and output PSNR (SSIM) of particular edge-preserving smoothing algorithms seeking to understand in what regions, of noise levels, the dominant effect of the algorithm is denoising or image abstraction.

5.4.3.2 Experiment 4

In these experiments we take images acquired from digital cameras taken in low light environments. The noise process for such systems are non gaussian as the image acquisition process for modern digital camera systems is complex with multiple stages associated with different noise processes. For a visual comparison we experiment on the images of figure 5.10. Figures 5.10 (a) and (d) show the full image of the subjects whilst (c) and (d) are zoomed in versions so that the noisy artifacts are more visible. This is to ensure the visibility of the noisy artifacts. All of the images used in the visual experiments are better appreciated when viewed full size on a computer monitor. We then plot the original image verses the recovered images. Visual comparisons can be seen in figures 5.11 and 5.12. The experiments are carried out with the Bilateral filter (BF), non-local means algorithm (NLM), Power iteration denoising (PID) and PageRank image denoising (PR) algorithms. The parameters of the algorithms are hand tuned to produce visually pleasing results. The left half of each image shows the original noisy image whilst the right half is smoothed by the respective filter. From the results we can observer that:

(1) These algorithms achieve edge preserving smoothing with PID best preserving the clarity of the edges and the non-local means algorithm performing worst in this aspect.

(2) The non-local means algorithm is seen to remove noisy artifacts but introduces a significant level of blur into the image.

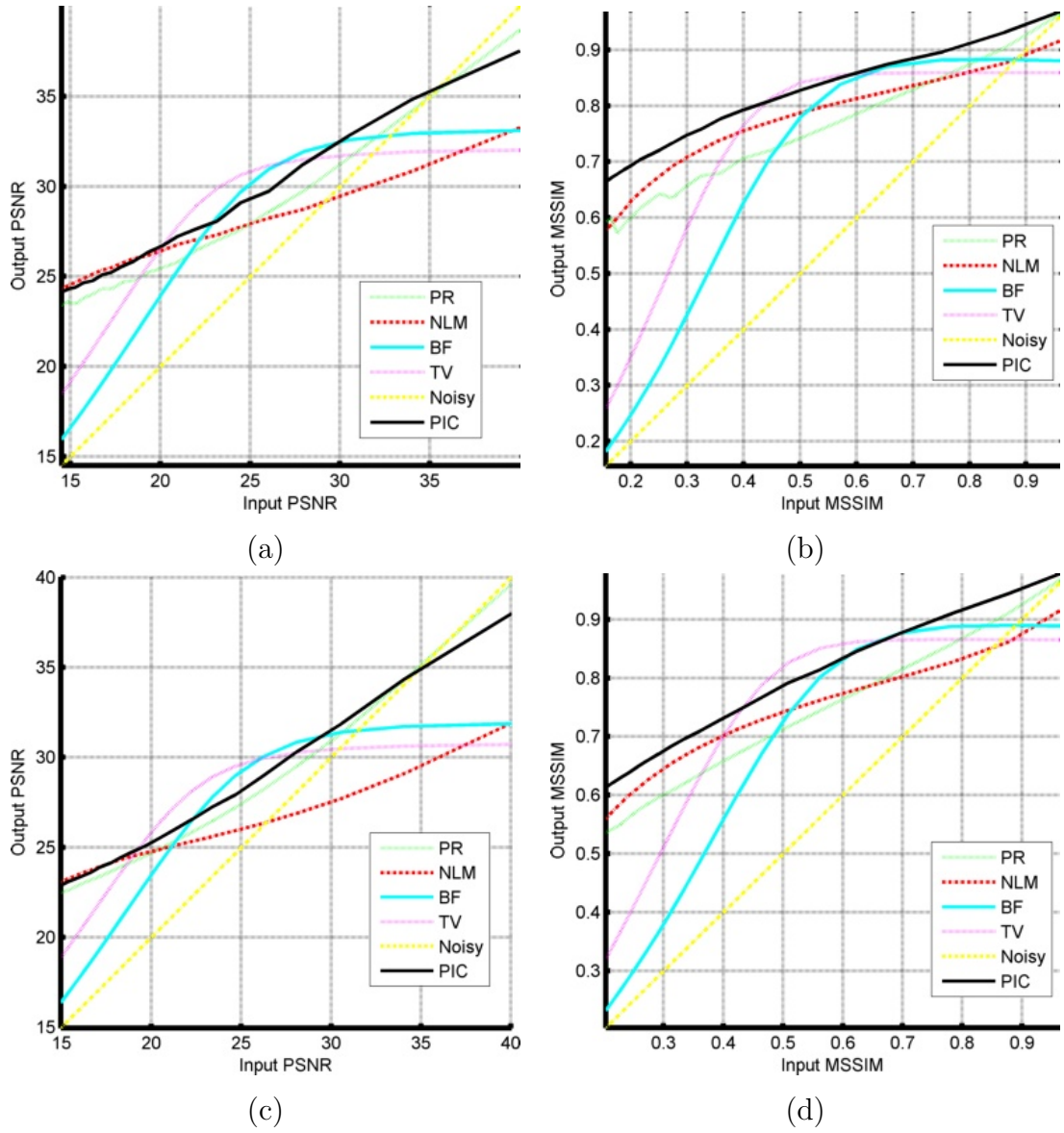


Figure 5.9: Comparisons of denoising performance of our methods PR and PID against the non-local means method, Bilateral filter and total variation method. The curves depict output PSNR as a function of input PSNR measured in dB. (a) and (b) Recovered PSNR and MSSIM scores for images of figure 5.1. (c) and (d) Recovered PSNR and MSSIM scores for images of figures 5.2 and 5.3.

(3) The PageRank denoising algorithm is seen to be the most "aggressive" smoothing algorithm producing "cartoon-like" characteristics to the image. Such characteristics are desirable in some applications in computational photography, particularly in non-photorealistic rendering and image abstraction [DeCarlo & Santella (2002),Winnemller *et al.* (2006)] .

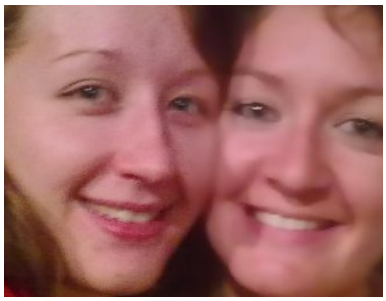
The visual aspects of these algorithms will be further investigated in the application to computational photography.

(a) (b)

(c) (d)

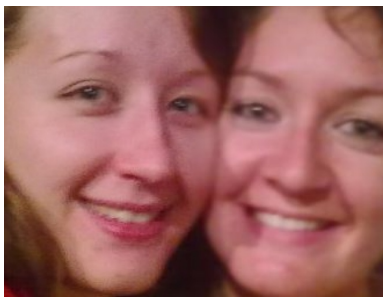
Figure 5.10: Digital camera images taken in low light environments showing noisy artifacts particularly evident on the skin of the main subjects. (a) and (c) are the full images whiles (b) and (d) are zoomed in versions so that the noisy artifacts are more evident.

(a) Bilateral Filter $\sigma_s = 3$, $\sigma_r = 0.1$



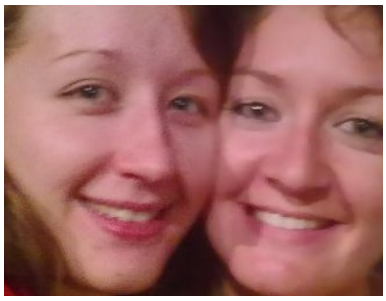
(b) Bilateral Filter $\sigma_s = 3$, $\sigma_r = 0.1$

(c) NLM $\sigma = 0.5$



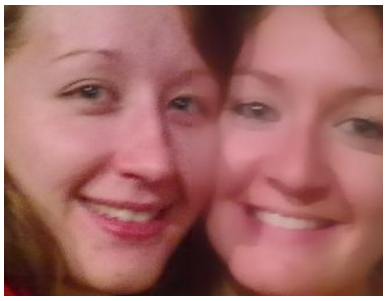
(d) NLM $\sigma = 0.5$

(d) PID $\sigma = 0.05$



(e) PID $\sigma = 0.05$

(e) PR $\sigma = 0.05$, $\Delta t = 0.95$



(f) PR $\sigma = 0.05$, $\Delta t = 0.95$

Figure 5.11: Visual comparisons taken in an outdoor scene. Right images are zoomed in versions of the left images. The left half of each image shows the original image whilst the right half is smoothed by the respective filter.

(a) Bilateral Filter $\sigma_s = 3$, $\sigma_r = 0.1$

(b) Bilateral Filter $\sigma_s = 3$, $\sigma_r = 0.1$

(c) NLM $\sigma = 0.5$

(d) NLM $\sigma = 0.5$

(d) PID $\sigma = 0.05$

(e) PID $\sigma = 0.05$

(e) PR $\sigma = 0.05$, $\Delta t = 0.95$

(f) PR $\sigma = 0.05$, $\Delta t = 0.95$

Figure 5.12: Visual comparisons of images taken in a low light environment. Right images are zoomed in versions of the left images. The left half of each image shows the original image whilst the right half is smoothed by the respective filter.

5.5 Discussion

Estimating a denoised image of a scene from a single noisy image is an under-constrained or ill-posed problem. Modern algorithms for denoising images acquired from digital cameras therefore estimate the denoised image from a set of noisy observations such as the system developed in [Petschnigg *et al.* (2004)]. These methods rely on edge-preserving smoothing algorithms as their main computational mechanisms. At a high level the aim of the edge-preserving smoothing algorithm is to

> *smooth noisy variations in the image whilst preserving meaningful structure.*

More verbosely we can establish five goals that we may wish to achieve when designing these algorithms

1. Perceptually flat regions should be as smooth as possible, that is noise should be completely removed from these regions.

2. Image boundaries should be well preserved implying that they should neither be blurred nor sharpened.

3. Texture detail should be preserved. This is possibly the hardest objective to achieve.

4. The global contrast of the image should be preserved.

5. No artifacts should be introduced as a result of the edge-preserving smoothing process.

In this chapter we presented two novel edge-preserving smoothing algorithms namely the PageRank denoising algorithm and the Power iteration denoising algorithm and experimentally compared them to other related state-of-the art edge preserving smoothing algorithms namely the Bilateral filter, and non-local means filtering.
These novel algorithms can be derived from a probabilistic framework using conditional random field models. The random walk graph Laplacian being at the

heart of their computational framework. Applying these methods can be seen to be solving the Laplace and Poisson equations on a graph. As such we presented analytical comparisons with other graph based diffusion processes such as that proposed in [Elmoataz *et al.* (2008)]. We found that our algorithms are better suited to the edge preserving smoothing problem than the methods proposed in [Elmoataz *et al.* (2008)] as they do not require multiple heuristic parameters to achieve stable iterations.

From the experimental evaluations we find that our methods are good edge-preserving smoothing algorithms. They produce visually pleasing reuslts when applied to real images. To understand the utility of these novel algorithms in the next section we apply them in the emerging field of computational photography.

Relation to other image denoising methods. Image denoising is still a very active area of research in the image processing [Chatterjee & Milanfar (2010)] community. In the past three decades there have been a variety of denoising methods that have been developed in the image processing community. Although the methods are seemingly different they share a common goal: to keep meaningful edges and detail whilst removing noise: this paradigm is often referred to as *edge-preserving smoothing*.

Anisotropic diffusion methods [Perona & Malik (1990b),Weickert *et al.* (1998),Chambolle (2004a)] seek to effect the smoothing by a combination of image intensities in a localized neighborhood. The iterative update is generated by the discretization of a non-linear partial differential equation. These methods often suffer from over smoothing of image regions, instabilities due to discretization errors and computational expense. Our algorithms are analogous to these methods as we form diffusion processes on graphs. We maintain the key notion of the Laplacian from the partial differential equation methods.

Other methods seek to directly model the marginal statistics of neighboring image pixels by the introduction of a probabilistic image prior Roth & Black (2009); Zhu *et al.* (1998). These methods are formulated as Markov random field models and Bayesian inference is required to produce the restored image. The parameters of the image priors have to be learned before applying the model, which can be challenging. We related our algorithms to these methods through energy-minimisation of Markov random fields and conditional random fields.

Recently a Graph based method was proposed for image denoising [Zhang & Hancock (2006)]. The denoising is produced by computing the eigenvalues of the unormalized graph Laplacian which are then used to solve the graph Heat equation. Computing the eigenvalues of the graph Laplacian can be costly. Our method is similar to this method as we are using eigenvalues of the graph Laplacian to effect our denoising. We explicitly require the random walk graph Laplacian where as in their work the normalized graph Laplacian is used.

Relation to semi-supervised methods. This repeated update at the heart of algorithms 4 and 7, particularly the multiplication $M_{rw}\mathbf{u}^n$, has been used extensively in semi-supervised classification methods where it is viewed as an iterative average or backward random walk. The update has been used extensively in graph based methods to propagate class information [Zhu *et al.* (2003a)].

If we consider an electric network built from our graph G using resistors with conductances[1] given by $\frac{w_{ij}}{\sum_j w_{ij}}$ between nodes u_i and u_j. Our objective is to compute the voltages of the nodes (that is the image intensities) for the network at steady state. Now if we denote the current between the nodes u_i and u_j by I_{ij} and the potential difference by $V_{ij} = u_i - u_j$ then we have Ohm's law given by

$$I_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} V_{ij} \tag{5.33}$$

and Kirchoff's current law at a node is given by

$$\sum_j I_{ij} = 0. \tag{5.34}$$

Kirchoff's current law tells us that the sum of the electrical currents flowing out of a node i is equal to the sum of the electrical currents flowing into node i. That is the electrical current in the network is conserved. Therefore we can write the expression

$$\sum_j \frac{w_{ij}}{\sum_j w_{ij}} (u_i - u_j) = 0. \tag{5.35}$$

Finally it is not hard to show that

$$\sum_j \frac{w_{ij}}{\sum_j w_{ij}} (u_i - u_j) = u_i - \sum_j \frac{w_{ij}}{\sum_j w_{ij}} u_j \tag{5.36}$$

[1] We recall that the conductance is the inverse of resistance.

which is just the action of the random walk Laplacian on a node

$$L_{rw}\mathbf{u} = 0. \tag{5.37}$$

5.6 Applications to Computational Photography

Generally photographic principles have remained largely unchanged since the invention of the camera by Joseph Nicéphore in the 1820s [Levoy (2010)]. In optical photography an optical lens focuses light from a scene onto a photographic plate which records the scene radiance information directly to form a picture. As the picture is a simple copy of the optical information reaching the plate improvements in picture quality have been largely achieved by refining the optics, recording methods and the process of developing the image. Recent refinements have been dramatic due to the switch over from photographic film sensors to digital image technology.

Computational photography challenges the traditional view of photography by considering the image the sensor captures to be an intermediary stage in a wider computational pipeline to generate the desired picture. Broadly speaking computational photography, in recent parlance [Levoy (2010)], refers to sensing strategies and algorithmic techniques that enhance or extend the capabilities of the digital photography process. It brings together aspects of computer graphics, computer vision and image processing with the main goal of this area being to redefine the camera by using computational techniques to produce a new level of images and visual representation that enhance or extend the capabilities of the modern digital camera. Often it requires multiple images which are then combined in some way to produce the final image. Many computational photography techniques take the form

> "*capture a burst of images* $\boldsymbol{f}_1, \boldsymbol{f}_2 \ldots \boldsymbol{f}_n$ *varying capture setting* x *and combine them to produce a single enhanced image* $\boldsymbol{u}$ *that exhibits better property* y*"*.

Representative techniques include high dynamic range imaging (HDR) [Debevec & Malik (1997),Mitsunaga & Nayar (2000),Durand & Dorsey (2002)], flash/no-flash imaging [Petschnigg *et al.* (2004),Eisemann & Durand (2004)], transfer of photographic look [Bae *et al.* (2006)], photography under structured lighting [Fattal *et al.* (2007)] and image editing [Oh *et al.* (2001),Khan *et al.* (2006)].

5.6.1 Prior Work

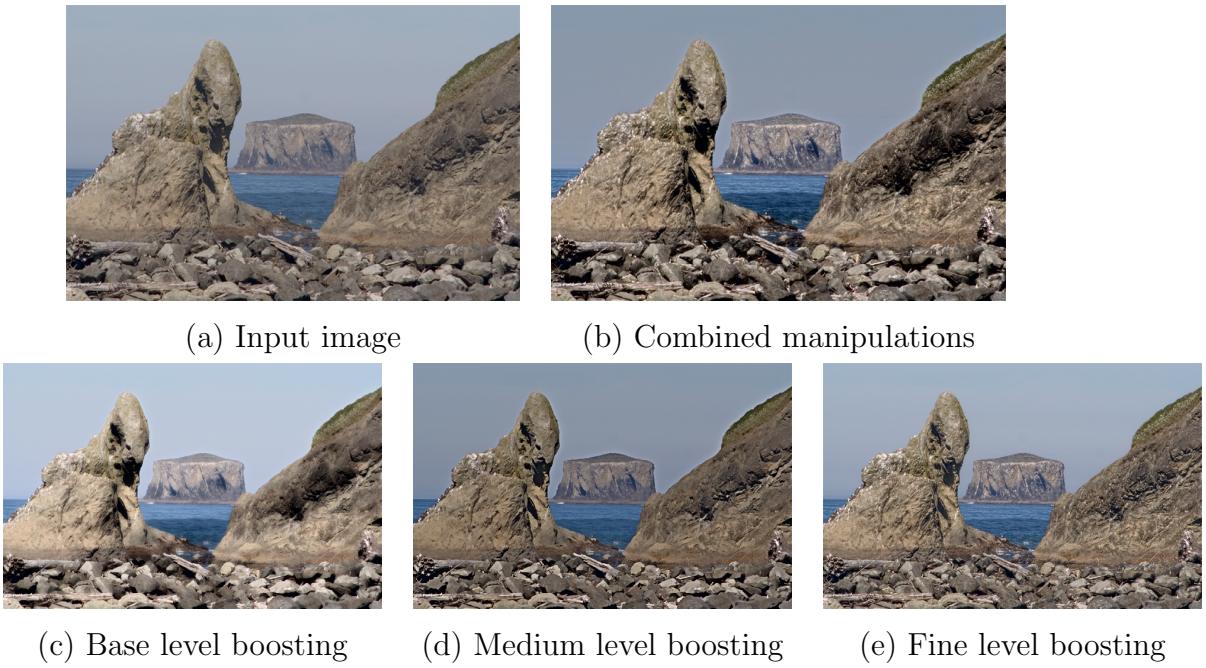
HDR Imaging. Real world scenes contain a high level of brightness variation that most commercially available digital cameras cannot capture. Scene, radiance, for instance, may contain 4 orders of magnitude from dark shadow regions to fully lit bright regions where by typical CCD and CMOS imaging sensors may only capture about 2^8 or 2^{12} levels [Kang *et al.* (2003)]. This limited range of imaging sensors has inspired many solutions over recent years for capturing the high dynamic range of scenes using conventional camera technology. Popular methods are based upon combining multiple images of a scene acquired at different exposures [Mann & Picard (1995),Debevec & Malik (1997),Mitsunaga & Nayar (2000)].
Creating a viewable high dynamic range image requires 3 stages:

1. estimation of a radiometric response function from aligned images
2. estimation of a radiance map by selection/blending of pixels at different exposures
3. tone-mapping the high dynamic range image into a displayable image.

Once the radiance map is obtained it is necessary to display the images on a lower dynamic range display such as a computer screen or in print media. This is the tone mapping problem. Recent tone mapping algorithms [Tumblin & Hodgins (1999),Durand & Dorsey (2002),Farbman *et al.* (2008)] require the decomposition of an image into a piecewise smooth base layer containing large scale intensity variations and a residual base layer capturing smaller scale fine details. Edge-preserving smoothing algorithms are utilised to effect such tasks of which the Bilateral filter has emerged as the de facto method.
Flash/no-flash Imaging. [Eisemann & Durand (2004),Petschnigg *et al.* (2004)] developed techniques for noise removal and detail transfer from flash/no-flash image pairs acquired from digital cameras. The technique of [Eisemann & Durand (2004)] enhances photographs shot in low-light environments by combining an image taken with the available ambient light and one taken with a spatially invariant flash. These two images are combined to form an enhanced image that

preserves the ambience of the original low-light environment and inserts detail and sharpness from the flash image. Similarly the technique of [Petschnigg *et al.* (2004)] combines flash/no-flash image pairs taken in low-light environments and combines these two images to transfer detail, generate denoised images, perform white balancing and red-eye reduction. The main computational mechanisms at the heart of these techniques is the Bilateral filter. Closely related to these techniques is the work of [Bae *et al.* (2006)] that utilises the Bilateral filter in a multi-scale strategy for the management of tone and photographic look. Their technique can transfer the "photographic look" from one image to another.

Multi-scale Methods. In these applications it is desirable to have control over the detail and tone in the image. Further to this one may also desire to be able to operate on details at a variety of scales. A photographer, for example, may desire to separately manipulate the tone at several scales in order to add depth and increase detail clarity. The resulting operations may recombine these manipulations. Figure 5.13 shows an example of multi-scale tone and detail manipulation using our PageRank denoising algorithm. Traditionally operations on images at multiple scales were effected by the usage of Laplacian pyramids. Unfortunately the usage of Laplacian pyramids is known to produce "halo" and "ringing" effects around edges in the manipulated images [Farbman *et al.* (2008)]. Traditionally these Laplacian pyramid methods used linear isotropic smoothing filters. To reduce the haloing and ringing artifacts anisotropic diffusion filters such as the Perona-Malik type [Perona & Malik (1990b)] and the Bilateral filter [Tomasi & Manduchi (1998)] were introduced to the processing chain. The Bilateral filter later emerged as the de facto filter of choice in such applications. As was shown in [Farbman *et al.* (2008)] the Bilateral filter is well suited for noise removal and extraction of detail at fine scales but is less effective for extraction of details at arbitrary scales which is a necessary requirement for multi-scale decompositions; hence they advocated and demonstrated the weighted least squares framework as a superior method for such applications.



(a) Input image (b) Combined manipulations

(c) Base level boosting (d) Medium level boosting (e) Fine level boosting

Figure 5.13: Multi-scale tone and detail manupulations. (a) Input image. (b) Output image as a result of combining detail manipulations at three scales. (c), (d) and (e) example boosting of details at the base (coarse) level, medium level and fine levels respectively.

5.6.2 Overview

These recently proposed computational photography techniques require the decomposition of an image into a piecewise constant smooth base layer containing large scale intensity variations and a residual detail layer capturing smaller scale fine details. In such decompositions it is often desirable to control the spatial scale of extracted details and to be able to manipulate details at multiple scales whilst avoiding visual artifacts. Here we propose a new way to construct multi-scale image decompositions by advocating the usage of the edge-preserving smoothing filters developed in this chapter; namely the PID and PageRank denoising filters. With our method it is easy to produce multi-scale image representations by controlling just one parameter; the width parameter of a Gaussian potential function. We compare our results to decompositions produced by the recently introduced weighted least squares edge-preserving filter [Farbman *et al.* (2008)] which was shown to be better suited for progressive coarsening of images and multi-scale detail extraction than the Bilateral filter. The Bilateral filter is the de facto method of choice in computational photography applications. We show how to construct edge-preserving multi-scale decompositions and apply them to the computational photography problems such as multi-scale tone and detail enhancement and flash/no-flash photography.

5.6.2.1 Methodology

In computational photography tasks images often are decomposed into a base layer and a detail layer. The base layer is generated by an application of an edge-preserving filter to the original layer (sometimes to the log of the luminance channel in the CIELAB colour space representation). The detail layer is then generated as a difference (or quotient) between the original image and the base layer. The resulting layers may be manipulated separately and recombined to affect the desired computational photography objective. In the reduction of dynamic range in HDR images, for example, the base layer can be subject to a non-linear compressive mapping whilst the detail layer is subject to boosting or attenuation [Pattanaik *et al.* (1998),Tumblin & Hodgins (1999),Durand & Dorsey (2002)]. In image stylisaition and abstraction [DeCarlo & Santella (2002),Winnemller *et al.*

(2006)] detail layers are discarded whilst the base layers are further processed to produce a stylized look. With our methods it is easy to produce multi-scale image representations by controlling just one parameter; the width parameter of the Gaussians that generate the edge weights for the PID and PageRank denoising algorithms.
In general multi-scale decompositions can be generated as follows: we denote by $\mathbf{f}$ the input image for which we would like a $(k+1)$-level decomposition. We denote by

$$\mathbf{u}^1, \ldots, \mathbf{u}^k \tag{5.38}$$

progressive coarsenings of the original image $\mathbf{f}$ and the coarsest version, $\mathbf{u}^k$, is called the base layer, $\mathbf{b}$. We then have k detail levels defined as

$$\mathbf{d}^i = \mathbf{u}^{i-1} - \mathbf{u}^i. \tag{5.39}$$

The original image can easily be shown to be given by

$$\mathbf{f} = \mathbf{b} + \sum_{i=1}^{k} \mathbf{d}^i \tag{5.40}$$

A coarsening sequence $\mathbf{u}^1, \ldots, \mathbf{u}^k$ can be generated as follows: We apply an edge-preserving filter algorithm to the original image k times, at each iteration varying the values of the parameters controlling the characteristics of the edge-preserving filter.

$$\begin{aligned} \mathbf{u}^n &\leftarrow F^n_{edge}(\mathbf{f}) \\ \text{for } n &= i, \ldots k \end{aligned} \tag{5.41}$$

where F_{edge} is the edge-preserving smoothing algorithm and $\mathbf{f}$ is the base layer.

5.6.3 Comparisons

In these experiments and comparisons we aim to understand the ability of the proposed edge-preserving filters to achieve progressive coarsening. We provide comparisons with the Bilateral filter and the weighted least-squares filter [Farbman *et al.* (2008)]. Firstly we recall that the Bilateral filter is a simple nonlinear

filter where each pixel in the filtered result is assigned a weighted combination of pixel neighbours. The weightings decrease with both spatial distance and increasing difference in intensity values. Formally the Bilateral filter can be expressed as the sum

$$u_i = \frac{1}{K_i} \sum_j G_{\sigma_s}(|i-j|) G_{\sigma s}(|f_j - f_i|) f_i \tag{5.42}$$

$$K_j = \sum_j G_{\sigma_s}(|i-j|) G_{\sigma s}(|f_j - f_i|) \tag{5.43}$$

where $\mathbf{f}$ is an image and the subscripts i and j indicate spatial locations of pixels. Kernel functions G_{σ_s} and G_{σ_r} are chosen to be Gaussians where σ_s determines the spatial support of the Bilateral filter and σ_r controls the sensitivity of the filter to edges.

As way of illustration we study the application of the Bilateral filter to the Strangers image, figure 5.14 (a). The Strangers image is roughly piecewise constant with step edges of varying magnitudes. The goal of the edge-preserving smoothing is to achieve a similar degree of smoothing in the piecewise smooth regions whilst preserving the edge transitions. Figure 5.14 demonstrates a phenomenon that has been noted by other authors [Farbman *et al.* (2008),Tomasi & Manduchi (1998)] that more aggressive smoothing by the Bilateral filter cannot be achieved by only increasing the spatial support parameter, σ_s, but the range support parameter σ_r must be increased in tandem. However this strategy reduces the ability of the Bilateral filter to preserve edges as is evident in figure 5.14 (d) and (e). While the Bilateral filter is effective at soothing small intensity changes whilst preserving edges, its ability to achieve progressive coarsening is rather limited. Other authors have noted that it is possible to achieve more aggressive smoothing with the Bilateral filter by applying it in an iterative fashion [Winnemller *et al.* (2006),Fattal *et al.* (2007)], however such a strategy is known to over sharpen edges producing results resembling the mean-shift filter [Comaniciu & Meer (2002),Paris & Durand (2008)]. This is undesirable as an ideal edge-preserving smoother must not sharpen edges as this may introduce halo and ringing artifacts when used in applications. This demonstrates that the Bilateral filter has a trade off between the ability to preserve edges and increase

smoothing hence making it difficult to control in applications requiring multi-scale decompositions.

As an alternative to the Bilateral filter [Farbman *et al.* (2008)] proposed the weighted least-squares filter (WLS) as an edge-preserving filter. The weighted least-squares filter can formally be introduced in the following optimisation framework: We seek the image $\mathbf{u}$ that minimises the functional

$$\sum_i \lambda \left[a_x(f_i) \left(\frac{\partial u_i}{\partial x} \right)^2 + a_y(f_i) \left(\frac{\partial u_i}{\partial y} \right)^2 \right] + \sum_i (f_i - u_i)^2 \tag{5.44}$$

where the second term of this equation is the data term whilst the first term is a smoothness term that seeks to achieve smoothness by minimising the local variation of the recovered image $\mathbf{u}$ through a weighted combination of the partial derivatives. The weights are chosen as follows

$$a_x(f_i) = \left(\left| \frac{\partial l_i}{\partial x} \right|^\alpha + \epsilon \right)^{-1} \quad a_y(f_i) = \left(\left| \frac{\partial l_i}{\partial y} \right|^\alpha + \epsilon \right)^{-1} \tag{5.45}$$

which gives rise to the operator for interactive local tone adjustment proposed in Lischinski *et al.* (2006). The parameter λ balances the trade off between the data fit and the smoothness whilst α (typically between 1.2 and 2.0) determines sensitivity to gradients. Where l is the log-luminance channel of the input image and ϵ is a small constant (typically 0.0001) that prevents division by 0 in regions of constant image intensity.

Essentially finding the minimum of equation 5.44 can be thought of as solving an anisotropic partial differential equation based image filter. The anisotropic diffusion is determined by the multipliers of the image derivatives. Equation 5.44 can be interpreted using the variational calculus by seeking an image u that minimises the functional

$$\lambda \int a_x(f) \left(\frac{\partial u}{\partial x} \right)^2 + a_y(f) \left(\frac{\partial u}{\partial y} \right)^2 d\Omega + \int (f - u) d\Omega \tag{5.46}$$

where Ω is the image domain. Minimising this functional results in the partial differential differential equation (Euler-Lagrange equations)

$$\nabla \cdot (a(|\nabla f|) \cdot \nabla u) + (f - u) = 0. \tag{5.47}$$

This is the form proposed by [Alvarez & Morel (1994b)] where they proposed Gaussian potentials for their choice of the function a that generates the anisotropic effect. These formulations require suitable choices of boundary conditions and a finite difference scheme to cast the partial differential equations into a discrete form. Furthermore it may be subject to stability conditions in its numerical simulation as is the case with partial differential equation based methods [Morton & Mayers (2005)] when applied to image processing [Sethian (1996),Osher & Fedkiw (2003)]. Moreover this method requires the choice of two parameters, α and λ, which have to be related to the scale of the edge-preserving smoother. In comparison to our Laplacian operator the parameter σ is interpreted directly as a scale parameter in our framework. It controls the the width of the Gaussian potentials expressing the similarity between pixels. The parameter Δt is chosen to be close to 1 so that the PageRank filter algorithm is a solution of the Poisson equation on a graph.

How does the mathematics translate to visualisations? In figure 5.16 we note with the weighted least-squares filter PageRank denoising and Power Iteration denoising we can increase the smoothing without significantly compromising the the edge preserving nature of these filters. This is in direct contrast to the Bilateral filter algorithm. Moreover we note that our methods, PageRank denoising and Power Iteration denoising, are controlled by only one parameter σ, which we interpreted as the scale parameter in the graph construction. This makes this method attractive as it is easy to control. Coarsening sequences generated by the weighted least squares filter, PageRank denoising and Power Iteration denoising are shown in figure 5.2. The left column shows a sequence generated by the weighted least-squares algorithm [Farbman *et al.* (2008)], middle column using the Power Iteration denoising algorithm and the right column using the PageRank denoising algorithm for various values of σ. In the PageRank filter algorithm we fix $\Delta t = 0.95$.

5.6.4 Applications and Results

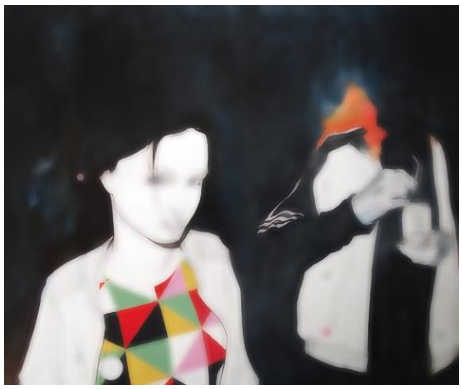
Here we demonstrate the efficiency of our edge-preserving algorithms by applying them to computational photography tasks that require multi-scale decompositions

(a) input image

(b) Bilateral Filter $\sigma_s = 4$, $\sigma_r = 0.3$

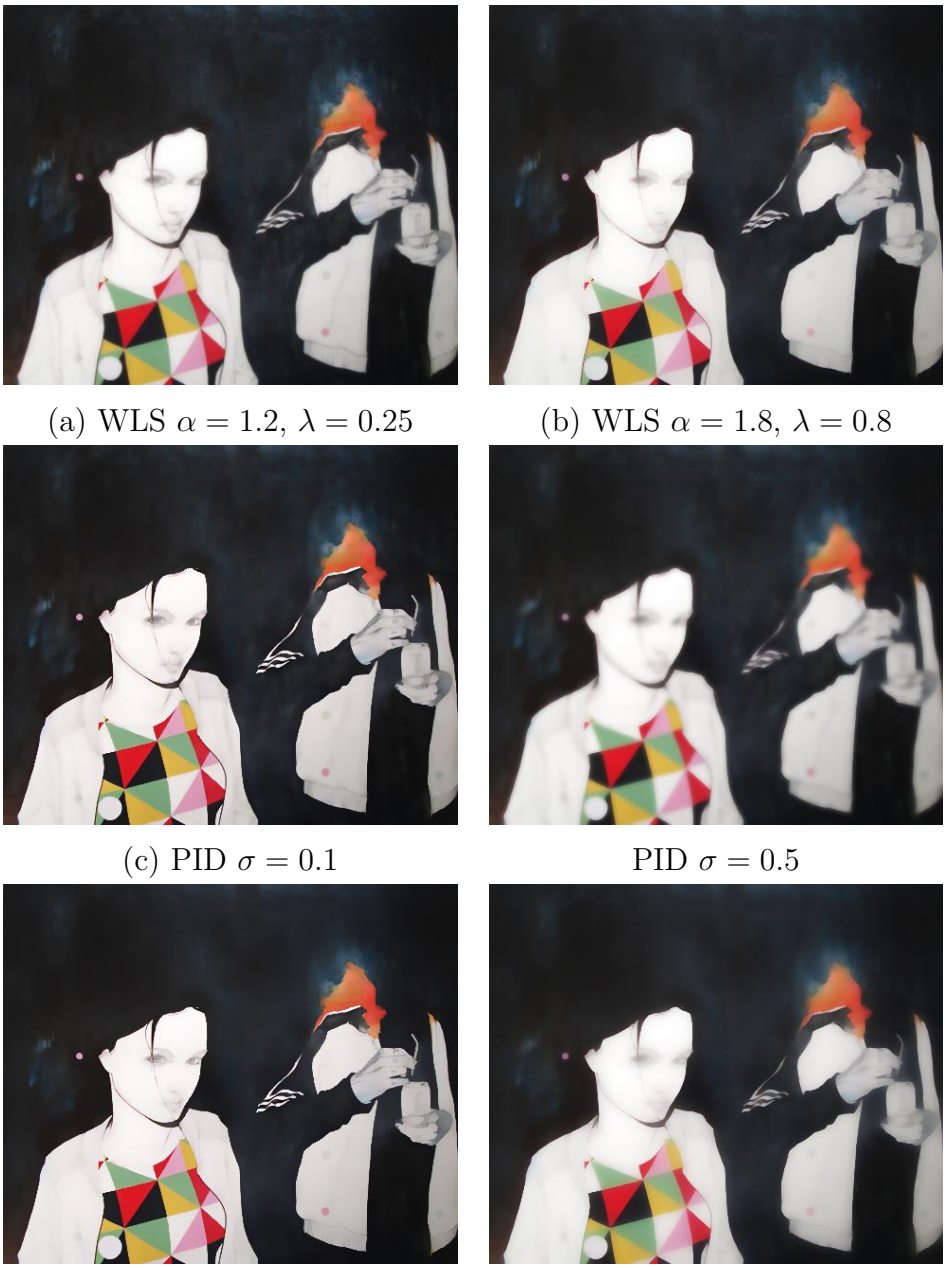
(c) Bilateral Filter $\sigma_s = 12$, $\sigma_r = 0.15$

(d) Bilateral Filter $\sigma_s = 12$, $\sigma_r = 0.3$



(e) Bilateral Filter $\sigma_s = 12$, $\sigma_r = 0.45$

Figure 5.14: (a) Input image: Strangers © courtesy of Kombo Chapfika *http://www.kombochapfika.com*. (b) - (e) Progressive filtering of Strangers image with the Bilateral filter.

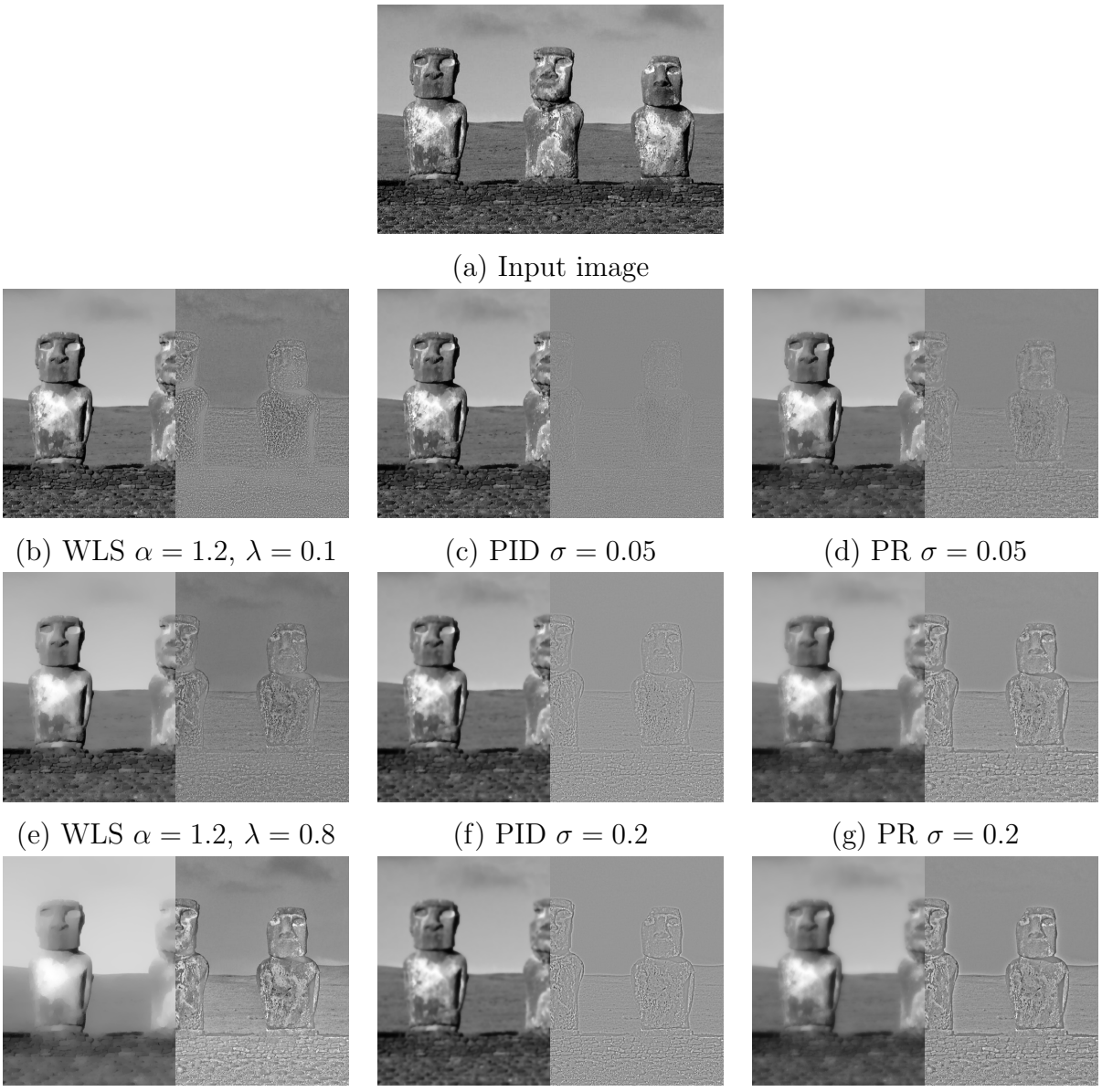


(a) WLS $\alpha = 1.2$, $\lambda = 0.25$ (b) WLS $\alpha = 1.8$, $\lambda = 0.8$

(c) PID $\sigma = 0.1$ PID $\sigma = 0.5$

(d) PR $\sigma = 0.05$ (e) PR $\sigma = 0.2$

Figure 5.15: Filtering of the Strangers image with a variety of filters: (a) and (b) weighted least-squares filtering [Farbman *et al.* (2008)], (c) and (d) Power Iteration denoising, (d) and (e) PageRank denoising $\Delta t = 0.95$.



(a) Input image

(b) WLS $\alpha = 1.2$, $\lambda = 0.1$ (c) PID $\sigma = 0.05$ (d) PR $\sigma = 0.05$

(e) WLS $\alpha = 1.2$, $\lambda = 0.8$ (f) PID $\sigma = 0.2$ (g) PR $\sigma = 0.2$

(h) WLS $\alpha = 1.2$, $\lambda = 6.4$ (i) PID $\sigma = 0.5$ (j) PR $\sigma = 0.5$

Figure 5.16: Multi-scale image decompositions. Left column: three levels computed using the weighted least-squares (WLS) method [Farbman *et al.* (2008)]. Middle images: levels computed using Power Iteration denoising (PID) for various values of σ and right image: levels computed using PageRank denoising (PR) for various values of σ and $\Delta t = 0.95$. The left half of each image shows the coarsening, while the right half visualises the corresponding detail layer. The spatial scale of the details increases from one level to the next.

of an image. A set of simple tools can be implemented for the tasks of multi-scale tone and detail manipulation and denoising with detail transfer using flash/no-flash image pairs.

5.6.4.1 Multi-scale tone and detail manipulation

A simple tool for interactive tone manipulation at multiple scales can be constructed as follows: Given an image $\mathbf{f}$ we can construct a three-level decomposition of the CIELAB lightness channel such that we obtain a base layer $\mathbf{b}$ and two detail layers $\mathbf{d}^1$ and $\mathbf{d}^2$ using equation 5.41. The manipulations can be effected pixel-wise using the equation

$$g_i = \mu + S(\delta_0, \eta b_i - \mu) + S(\delta_1, d_i^1) + S(\delta_2, d_i^2) \tag{5.48}$$

proposed in [Farbman *et al.* (2008)], where g_i is the result of the nonlinear combination, μ is the mean of the lightness channel in the CIELAB representation and S is a sigmoid curve

$$S(a, x) = \frac{1}{1 + \exp(-ax)}. \tag{5.49}$$

The term $S(\delta_0, \eta b_i - \mu)$ controls the per-pixel exposure and contrast of the base layer. The parameter η controls the exposure whilst δ_0 is a boosting factor. The remaining terms control the boosting of the medium and fine scale details with δ_1 and δ_2 as boosting factors.
This simple tool is effective for manipulation of tone, contrast and detail at multiple scale. It requires rather extreme manipulations for artifacts to appear in the resultant combined images. Figures 5.13 and 5.17 show example manipulations. For these results we used the Power Iteration denoising algorithm with $\sigma = 0.1$ for fine scale filtering and $\sigma = 0.2$ for coarse scale filtering. We used the PageRank denoising algorithm with $\Delta t = 0.95$, $\sigma = 0.05$ for fine scale filtering and $= 0.95$, $\sigma = 0.2$ for coarse scale filtering. For the weighted least-squares filter, as in [Farbman *et al.* (2008)], we use $\alpha = 1.2$, $\lambda = 0.1$ for fine scale filtering and $\alpha = 1.4$, $\lambda = 0.4$ for coarse scale filtering. We find that with our method we can produce similar manipulations to those proposed in [Farbman *et al.* (2008)].

(a) Input image

(b) Base level (WLS)

(c) Base level (PID)

(d) Base level (PR)

(e) Medium level (WLS)

(f) Medium level (PID)

(g) Medium level (PR)

(h) Fine level (WLS)

(i) Fine level (PID)

(j) Fine level (PR)

(h) Combined (WLS)

(i) Combined (PID)

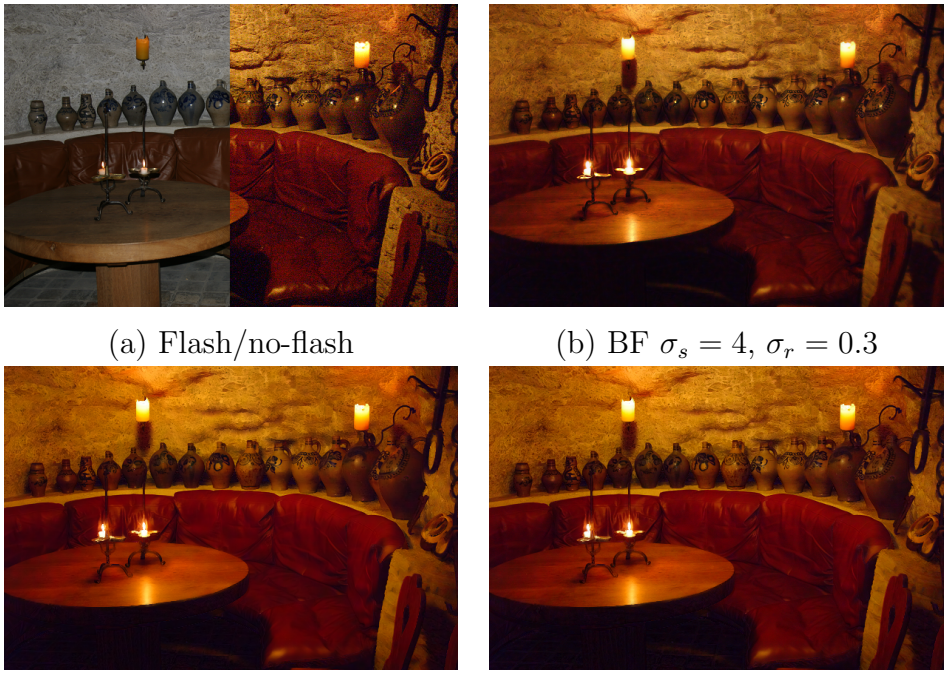
(j) Combined (PR)

Figure 5.17: Multi-scale tone and detail manipulation using weighted least-squares (WLS), Power Iteration denoising (PID) and PageRank (PR) denois.

5.6.5 Denoising with detail transfer

In low light environments a photographer is usually faced with an option of taking a photograph with a flash or without a flash (no-flash photographs). An image captured with a flash may cause distracting artifacts such as sharp shadows at silhouettes and flat and harsh lighting. Alternatively the photographer may choose to capture a no-flash image. This too poses some challenges. When using the available light the photographer may use a long exposure time so that the camera can collect enough light to produce a visible image. However this strategy may result in motion blur. Alternatively the photographer may increase the camera gain, using the ISO setting, and reduce the exposure time. This strategy is known to increase the noise in the resulting image. [Eisemann & Durand (2004),Petschnigg *et al.* (2004)] proposed to combine a flash image and a no-flash image of the scene and to use the desirable attributes of the acquired images. The flash image has a higher signal-to-noise ratio while the no-flash image retains the overall ambience and mood of the scene. Their techniques construct a Bilateral filter from the flash-image. The filter is then applied to the no-flash image to remove noise. The flash image is then decomposed into a base layer and a detail layer. The detail layer is then combined with the filtered no-flash image thereby enhancing the quality of the image captured in a low-light environment. These image pairs have to be captured in a relatively short period of time. It is appreciated that most consumer grade digital cameras do not have the functionality of capturing bursts of images in succession but it is envisaged that programmable platforms such as those developed in [Levoy (2010)] will allow developers to implement such tasks amongst others.

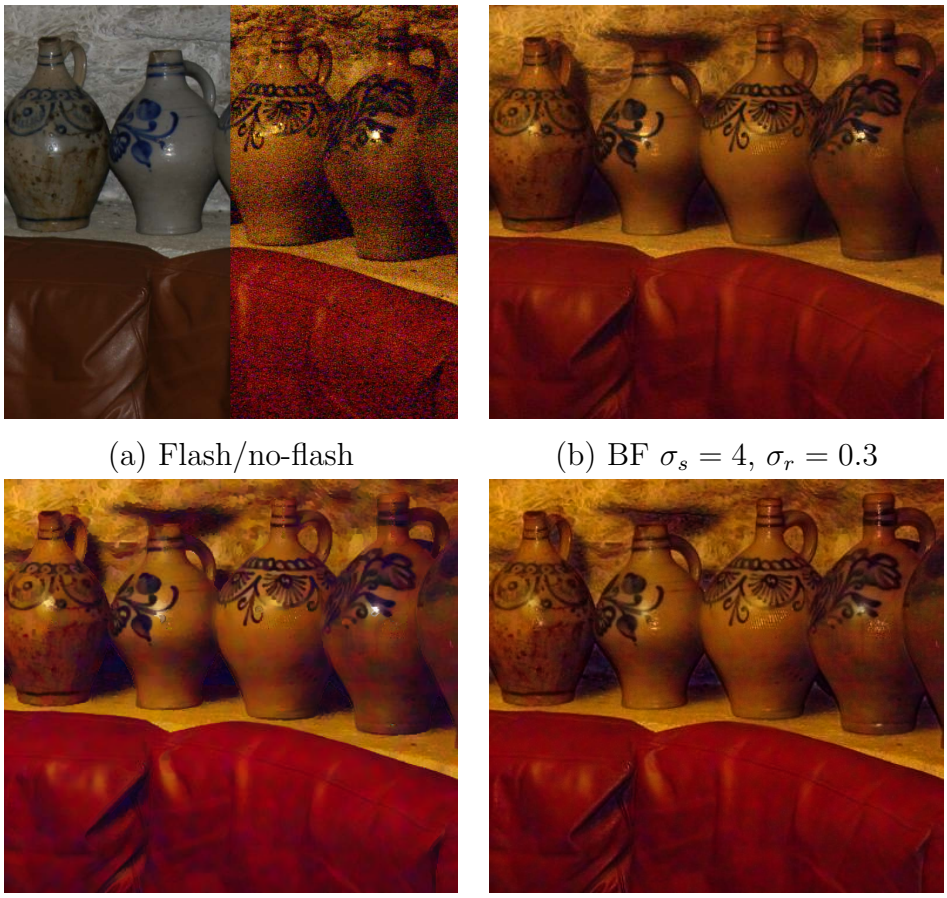
Building on these ideas we propose to use the flash images to generate the Laplacian random walk matrix for the Power Iteration denoising and PageRank denoising algorithms. We use these filters to decompose the flash image into a base layer and detail layer. We then filter the no-flash image and transfer the detail layer from the flash image to the filtered no-flash image using equation 5.48. Figures 5.18 and 5.19 show the results of application of our technique to denoising and detail transfer for flash/no-flash photography. We provide our results along side those produced using the Bilateral filter [Petschnigg *et al.* (2004)].



(a) Flash/no-flash (b) BF $\sigma_s = 4$, $\sigma_r = 0.3$

(c) PID $\sigma = 0.02$ (d) PR $\sigma = 0.02$

Figure 5.18: A photographer wishes to capture the ambience of the low-light environment but captures a noisy no-flash image. The flash image is then captured which has a higher signal-to-noise ratio but looses the ambience of the scene. (a) Left half shows a section of the flash image, right half shows a section of the no-flash image. (b) Denoising with detail transfer using the Bilateral filter (BF). (c) Denoising with detail transfer using the Power Iteration denoising algorithm (PID) .(d) Denoising with detail transfer using the PageRank denoising algorithm (PR).



(a) Flash/no-flash (b) BF $\sigma_s = 4$, $\sigma_r = 0.3$

(c) PID $\sigma = 0.02$ (d) PR $\sigma = 0.02$

Figure 5.19: (a) Left half shows a section of the flash image, right half shows a section of the no-flash image. (b) Denoising with detail transfer using the Bilateral filter (BF). (c) Denoising with detail transfer using the Power Iteration denoising algorithm (PID) .(d) Denoising with detail transfer using the PageRank denoising algorithm (PR).

5.7 Discussion

Multi-scale image decompositions are valuable for the development of digital darkroom techniques. The ability to produce multi-scale decompositions with edge-preserving filters that use simple intuitive parameters is desirable. We have presented novel methods for constructing multi-scale image decompositions using the random walk graph Laplacian. The methods are simple to use with intuitive parameters and can be interpreted in terms of the Laplace and Poisson equations on a graph. The Poisson construction (PageRank denoising algorithm) can be solved efficiently using Krylov subspace methods on graphics processor units.
rom a theoretical point of view this is significant as it draws relationships with recently introduced methods from pattern recognition and machine learning. Moreover we believe this method can have an impact on other computational photography applications such as tone mapping [Durand & Dorsey (2002)], tone manipulation for photographic look [Bae *et al.* (2006)], image abstraction and stylisation [Winnemller *et al.* (2006)] and detail enhancement from multi-light image collections [Fattal *et al.* (2007)].
In future we would like to investigate more sophisticated machine learning techniques for selection of the scale parameter.

Chapter 6

Interactive Image Segmentation

6.1 Chapter Summary and Contributions

The goal of this chapter is to use the diffusion process, based on the random walk graph Laplacian, in a semi-supervised segmentation framework. We will then compare and relate the proposed methods to similar methods in the literature.

6.2 Introduction

Throughout this thesis we have been motivating the iterative update

$$\mathbf{u}^{n+1} \leftarrow c\mathbf{f} + (1-c)\mathbf{u}^n \tag{6.1}$$

as a diffusion process for denoising images. In prior chapters we have discussed the relationship with semi-supervised machine learning strategies and shown this diffusion process to be a discrete scale-space. Now we will use this iteration to diffuse label information through the image in order to perform semi-supervised image segmentation. We consider the binary segmentation problem where we would like to segment the region into two regions or classes $\{+1, -1\}$. The vector $\mathbf{u}^n$ becomes an indicator vector of class labels, M_{rw} is a diffusion process and the parameter c becomes a diagonal matrix of constraints indicating our level of belief in the labeled regions.

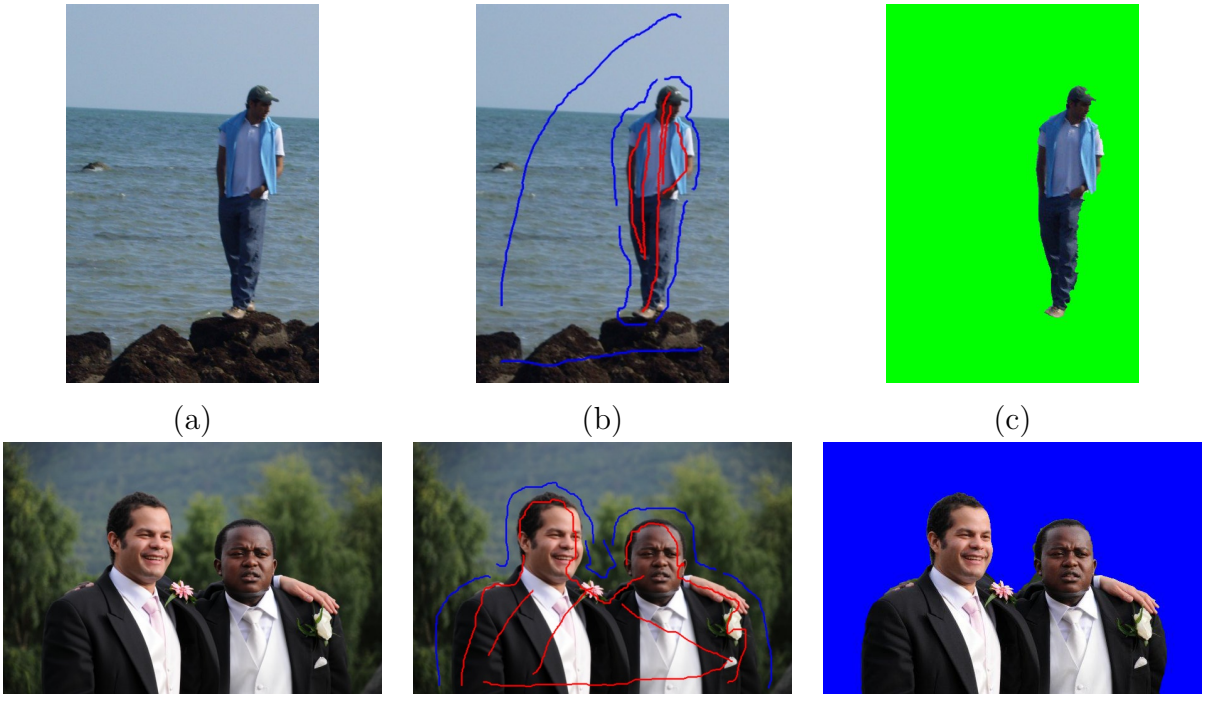


Figure 6.1: Semi-supervised image segmentation using the PageRank denoising algorithm. (a) and (d) Input images. (b) and (e) input images with user-supplied brush strokes. Red strokes indicate foreground region and blue strokes indicate background region. (c) and (f) the resulting image segmentations.

6.2.1 Semi-supervised segmentation

Interactive image segmentation is a problem of interest in the computer vision community. A user may want to extract a foreground object from an arbitrarily complex environment whose background may not be trivially extracted. As discussed in the literature survey on image segmentation such a framework arose from the inadequacies of fully automatic image segmentation frameworks to provide meaningful segmentations. In the interactive segmentation framework, which probably dates back to the intelligent scissors system developed in [Mortensen & Barrett (1995)], an oracle or domain expert is utilised to interactively provide control points to constrain the possible set of solutions that the optimisation framework may find hence producing more intuitive segmentations. This prob-

lem has obvious applications in image editing [Rother *et al.* (2004),Levin *et al.* (2008),Singaraju & Vidal (2008)] and medical image analysis [Grady *et al.* (2006)]. In the papers [Grady (2006)] and [Rother *et al.* (2004)], the authors motivate the desirable attributes of an interactive segmentaiton framework. This framework allows the user to give some guidance to the segmentation algorithms as to which are the regions of interest. These are given as a set of class labels for the semantically different regions of the image such as background and foreground regions. Figure 6.1 shows an example of interactive (semi-supervised) image segmentation using the PageRank denoising algorithm. Red brush strokes indicate foreground region and blue brush strokes indicate the background region.
In [Grady (2006)] the author defines a good interactive segmentation framework as one that allows the user the following

- fast computation
- fast editing
- ability to produce arbitrary segmentations with enough user interaction
- intuitive segmentations.

Likewise [Rother *et al.* (2004)] proposed that the aim of an interactive segmentation procedure is to achieve high performance at the cost of only a modest interactive effort on the part of the user.
Semi-supervised learning can be seen as a framework that may allow these objectives to be met. In this framework the user provides a small amount of labels on the data set. Labels, traditionally in the semi-supervised approach, may be time consuming to produce so we end up with a much larger set of unlabeled data points than labeled. In this framework we use our prior understanding of the geometric relationships between data points to propagate label information to the unlabeled nodes.

6.2.2 Algorithms

Given a graph $G = \{\mathbb{V}, \mathbb{E}\}$ with vertices labeled by 1 or -1 for two distinct classes and unknown classes labeled 0. Each node is allowed to diffuse label information

to neighboring nodes and this process is repeated until convergence. Pseudo code for the algorithm is given in algorithm 8. In the first algorithm the matrix C

Algorithm 8 PageRank Semi-supervised Segmentation (Hard Constraint)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W$
4: Initialise $\mathbf{f} \leftarrow (f_1, \ldots, f_l, 0, \ldots, 0)$, initialise C
5: **for** $n = 0, 1, 2, \ldots$ until convergence **do**
6: $\quad \mathbf{u}^{n+1} \leftarrow C\mathbf{f} + (1 - C)M_{rw}\mathbf{u}^n$
7: **end for**
8: classify nodes, $u_i \leftarrow 1$ if $u_i > 0$
9: classify nodes, $u_i \leftarrow -1$ if $u_i < 0$

contains along its leading diagonal, 1s for classified points and 0s for unclassified points. The rest of the entries of this matrix are zeros. The matrix of values C behaves as a hard constraint to enforce that at each iterate the initially labeled points $\{-1$ and $1\}$ maintain their values. The matrix M_{rw} then allows diffusion in regions of unlabeled data according to the update

$$u_i^{n+1} \leftarrow \sum_{u_i \sim u_j} \frac{w_{ij}}{\sum_j w_{ij}} u_j. \tag{6.2}$$

This equation can be seen to be a weighted average of the current labels of the neighbouring nodes. Thus at each iteration a node u_i receives a contribution from its j neighbouring nodes and an additional contribution given by its initial value. By relaxing the hard constraint of C a variant of this algorithm can be derived by treating C as a scalar value in the range $[0, 1]$. For the scalar constraint we will use lower case c. As will be shown in the evaluations this greatly speeds up the convergence of the algorithm. This relaxations gives us the standard fixed point iteration of the PageRank denoising algorithm.
If we recall the transformation

$$\Delta t = 1 - c \tag{6.3}$$

we can therefore motivate algorithm 9. An alternative approach would be to

Algorithm 9 PageRank Semi-supervised Segmentation (Soft Constraint)

1: Choose σ
2: Construct weight matrix W and degree matrix D
3: $M_{rw} = D^{-1}W$
4: Initialise $\mathbf{f} \leftarrow (f_1, \ldots, f_l, 0, \ldots, 0)$, initialise c
5: Initialise $\Delta t = 1 - c$
6: Solve $\mathbf{u} \leftarrow (1 - \Delta t)(I - \Delta t M_{rw}^{-1})\mathbf{f}$
7: classify nodes, $u_i \leftarrow 1$ if $u_i > 0$
8: classify nodes, $u_i \leftarrow -1$ if $u_i < 0$

directly minimise the harmonic energy

$$L_{rw}\mathbf{u} = \mathbf{f} \tag{6.4}$$

where $\mathbf{f}$ is the vector of labeled and unlabeled nodes and $\mathbf{u}$ are the class labels we seek to estimate. This can be achieved by partitioning the Laplacian matrix, L_{rw}, into blocks for labeled and unlabeled nodes

$$L_{rw} = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix} \tag{6.5}$$

and

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix}, \mathbf{u} = \begin{bmatrix} \mathbf{u}_l \\ \mathbf{u}_u \end{bmatrix} \tag{6.6}$$

where $\mathbf{f}_l$ denotes the a-prior user supplied class label and $\mathbf{f}_u$ denotes the mean values on the unlabeled data points. By imposing $\mathbf{u}_l = \mathbf{f}_l$ it was shown in [Zhu *et al.* (2003a)] that the harmonic solution is given by

$$\mathbf{u}_u = -L_{uu}^{-1} M_{lu} \mathbf{f}_l. \tag{6.7}$$

We can establish an equivalence between this method and our hard constrain algorithm 9. We can correspondingly write the PageRank update equation

$$\mathbf{u}^{n+1} \leftarrow C\mathbf{f} + (1 - C)M_{rw}\mathbf{u}^n \tag{6.8}$$

of the hard constrain algorithm 9 as

$$\begin{bmatrix} \mathbf{u}_l^{n+1} \\ \mathbf{u}_u^{n+1} \end{bmatrix} \leftarrow \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} M_{ll} & M_{lu} \\ M_{ul} & M_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{u}_l^n \\ \mathbf{u}_u^n \end{bmatrix} \tag{6.9}$$

By simple matrix arithmetic this system can be shown to reduce to

$$\begin{bmatrix} \mathbf{u}_l^{n+1} \\ \mathbf{u}_u^{n+1} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{f}_l^n \\ M_{ul}\mathbf{f}_l + M_{uu}\mathbf{u}_u^n \end{bmatrix}. \tag{6.10}$$

Therefore to find the unlabeled pixels we need to do the update

$$\mathbf{u}_u^{n+1} \leftarrow M_{ul}\mathbf{f}_l^n + M_{uu}\mathbf{u}_u^n. \tag{6.11}$$

Obviously at convergence it is assumed that $\mathbf{u}_u^{n+1} = \mathbf{u}_u^n$ therefore we need to solve the system

$$(I - M_{uu})\mathbf{u}_u = M_{ul}\mathbf{f}_l \tag{6.12}$$

and

$$\mathbf{u} = L_{rw}^{-1} M_{ul}\mathbf{f}_l \tag{6.13}$$

and the equivalence is established.

Unfortunately in our image processing problem L_{uu} is symmetric with row sums equal to zero. As a result the matrix has a zero determinant and is singular. Making the problem challenging to solve.

As can be observe the relaxed methods return a real valued vector function $\mathbf{f}$. We therefore require an appropriate classification rule. For class labels of $0, 1$ the Bayes classification rule is 0.5 and for class labels $\{-1, 1\}$ we use the Bayes classification rule of 0. This is our decision boundary. The algorithm can be run in a "one-vs-all" fashion where each node u_i is labeled with an $M-$dimensional vector where M is the number of classes. In the binary segmentation $M = 2$. The vector, $\mathbf{u}$, is 0 everywhere with a 1 for the class the labeled pixel belongs to. The segmentation algorithms are then run independently on each column vector $\mathbf{u}_m$ for $m = 1 \dots M$ and the pixels are assigned to their appropriate classes according to the classification rule

$$\mathbf{arg\ max}(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M). \tag{6.14}$$

6.2.2.1 Electric Network Analogy

In prior chapters we have provided analogous arguments with electric networks. Here we do the same. The analogy can provide us with some further insights and understanding of the algorithm. This argument has been developed in [Zhu *et al.* (2003a)] for semi-supervised machine learning. We present it here as it allows us a basis by which to compare with related algorithms.

If we consider an electric network built from our graph G using resistor with conductances[1] given by $\frac{w_{ij}}{\sum_j w_{ij}}$ between nodes u_i and u_j. We can connect the positively labeled nodes to a positive voltage source of $+1V$ and likewise we can connect the negatively labeled nodes to a negative voltage source of $-1V$. Our objective is to compute the voltages of the unlabeled nodes (that is their class labels). If we denote the current between the nodes u_i and u_j by I_{ij} and the potential difference by $V_{ij} = u_i - u_j$ then we have Ohm's law given by

$$I_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} V_{ij} \tag{6.15}$$

and Kirchoff's current law on an unlabeled node is given by

$$\sum_j I_{ij} = 0. \tag{6.16}$$

Kirchoff's current law tells us that the sum of the currents flowing out of a node i is equal to the sum of the electrical currents flowing into node i. That is the electrical current in the network is conserved. Therefore we can write the expression

$$\sum_j \frac{w_{ij}}{\sum_j w_{ij}} (u_i - u_j) = 0. \tag{6.17}$$

Finally it is not hard to show that

$$\sum_j \frac{w_{ij}}{\sum_j w_{ij}} (u_i - u_j) = u_i - \sum_j \frac{w_{ij}}{\sum_j w_{ij}} u_j \tag{6.18}$$

which is just the action of the random walk Laplacian on the labeled node

$$L_{rw}\mathbf{u} = 0. \tag{6.19}$$

[1] We recall that the conductance is the inverse of resistance.

6.3 Links with previous approaches

As already stated segmentation of natural images is a fundamental and challenging problem in image processing. In general there may be more than one interpretation of foreground/background objects of interest (in the absence of higher knowledge) making the problem ill-posed. The incorporation of prior information is then required to constrain the set of possible solutions. Recently, in the computer vision community this problem has been tackled by allowing a user to draw rough scribbles labeling the regions of interest. The image segmentation algorithms use this information to constrain the solutions. The problem can be expressed succinctly as that of finding the class label vector $\mathbf{u}$ that minimises the energy

$$E(\mathbf{u}) = \sum_{i \in \mathbb{V}} D_i(u_i) + \sum_{(i,j) \in n} V_{i,j}(u_i, u_j) \tag{6.20}$$

where $\mathbf{u} = (u_1, u_2, \ldots, u_N)^T$ is a set of binary labels for all nodes in the graph. D_i is a data association function operating on each pixel, $V_{i,j}$ is an interaction potential and n is a set of neighbouring pixels or clique. There have been a variety of algorithms introduced that follow on from this motivation [Boykov & Jolly (2001), Rother *et al.* (2004),Blake *et al.* (2004), Grady (2006),Bai & Sapiro (2007), Duchenne *et al.* (2008)]. Here we will review some closely related algorithms to our approach, particularly the graph cut approaches [Boykov & Jolly (2001), Blum *et al.* (2004)] and the random walker approach [Grady (2006)]. We will use the random walk approach as a benchmark for our algorithms.

6.3.1 Graph cuts

The graph cuts algorithm seeks to provide class labels by minimising the objective function

$$\mathbf{u}^T L_{un} \mathbf{u} + \sum_{i \in T} c_i (f_i - u_i)^2 \tag{6.21}$$

subject to the constraint

$$u_i = f_i \tag{6.22}$$

on the training set T. The training set is the labeled examples. This problem can be solved using a graph cut algorithm [Blum & Chawla (2001), Boykov & Jolly (2001), Blum *et al.* (2004), Boykov & Funka-Lea (2006)] using the Ford-Fulkerson min-cut max-flow method. The graph cut methods model the foreground/background pixels as sources/sinks and use the max-flow min-cut strategy to find a set of edges that separates with minimum total weights. The edge cuts are returned as the segmentation boundary. On the training set $c_i = \infty$. An experimental comparison of min-cut/max-flow algorithms for minimising the graph cut energy can be found in [Boykov & Komolgorv (2004)].

6.3.2 The Random Walker

Motivated by potential theory in electrical networks [Grady (2006)] proposed the random walker algorithm for image segmentation. They propose to propagate class label information to unlabeled regions using the harmonic function

$$\mathbf{u}_u = -L_{uu}^{-1} M_{lu} \mathbf{f}_l \tag{6.23}$$

where $\mathbf{u}_u$ is the unlabeled class information L_{uu} is the unnormalised Laplacian generated between unlabeled nodes and $\mathbf{f}_l$ is the class label information. Interestingly the authors give a random walk interpretation of their algorithm even though their method does not use the random walk Laplacian L_{rw} but instead uses the the unnormalised Laplacian L_{un}.

6.4 Experiments

To produce a quantitative evaluation of the algorithms developed in this chapter we appeal to the Microsoft GrabCut database[1] [Rother *et al.* (2004)]. This database contains a test set of 50 images of which each image contains a foreground object in a natural background environment. The objective of this dataset is to evaluate algorithms on the hard segmentation problem thus only objects with

[1] http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm

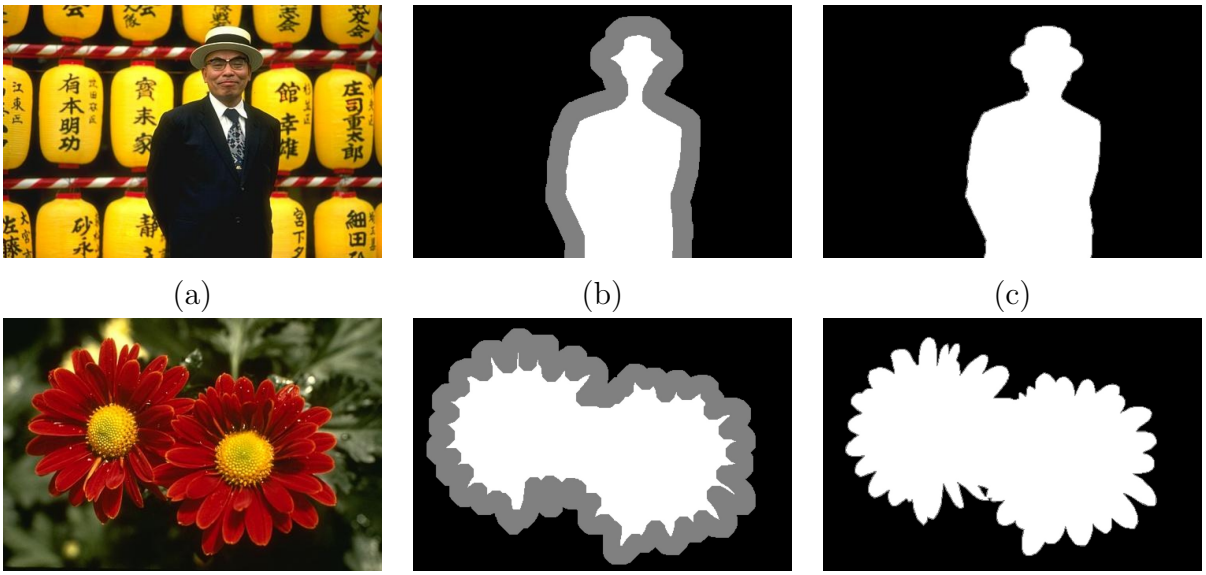
minimal (little or no) transparency are used. As a result partly transparent object such as trees, hair or glass are not included. In this database seeded images are provided by means of a trimap of which two kinds are assigned to each image. Type one is the user defined trimap as in figures 6.2 (b) and (e) and type two is the "expert trimap" obtained from tracing an object outline with a fine pen as in figures 6.2 (c) and (f). The pen trail covers pixels that could possibly be foreground or background pixels hence are excluded from the classification metric. The evaluation is computed as the segmentation error rate defined as

$$\epsilon = \frac{\text{no. misclassified pixels}}{\text{no. pixels in unclassified region}} \times 100. \tag{6.24}$$

We compare our method with the random walker algorithm [Grady (2006)] and two graph cut based methods: [Blake *et al.* (2004)] which formulates the graph cut algorithm as a probabilistic Gaussian mixture Markov random field (GMMRF) and uses a pseudo-likelihood algorithm for parameter learning. [Rother *et al.* (2004)] which uses an enhanced graph cut algorithm formulated as an "iterative estimation" of pixel labels.

6.4.1 Results

Figure 6.1 shows representative visual segmentations using the PageRank denoising algorithm. The associated linear system was solved using the conjugate gradient algorithm. Figures 6.3 and 6.4 show experimental evaluations comparing the random walk method (RW) with our methods based on Power Iteration denoising (PID), PageRank denosing (PR) and variants of the PageRank denoising algorithm. The variants of the PageRank denoising algorithm implies that the PageRank denoising algorithm is solved with one of the iterative solvers discussed in the previous chapter. In all our experiments the maximum allowable number of iterations is 1000. If the algorithm has not converged in the prescribed number of iterations it is terminated. Figure 6.3 shows the evaluation computed as the misclassification error rate (equation 6.24). The unit of error is percent. The error rate is reported for each image in the Microsoft GrabCut database. From this figure it is not hard to see that the random walk algorithm does not perform as well as the Power iteration denoising or PageRank denoising algorithms on



(a) (b) (c)

(d) (e) (f)

Figure 6.2: Two images from the MSRC GrabCut database. (b) and (e) User defined trimaps with foreground (white), background (black) and unclassified (grey) pixels. (c) and (f) Expert defined trimaps with classified foreground (white), background (black) and unclassified (grey) pixels; unknown pixels here refer to ambiguous pixels too close to the object boundary for the expert to classify.

this database. This is highlighted in table 6.1 where the average and median misclassification error rates are reported in tabular form for each algorithm. The bottom graph of figure 6.3 reports the misclassification rates for variants of the PageRank denoising algorithm evaluated over the Microsoft GrabCut database. Figure 6.4 reports the run times, in seconds, of the algorithms over the Microsoft GrabCut database. From these graphs we can see that the random walker algorithm is not as fast as the PID and PR algorithms when evaluated over this database. This is highlighted in table 6.2 where the average and median run times are reported.

Versus graph cuts. Table 6.3 shows comparisons with the graph cut based methods, GMMRF [Blake *et al.* (2004)] and GrabCut [Rother *et al.* (2004)], from which we see that our methods are competitive.

Parameter sensitivity. Figure 6.5 shows the images from the Microsoft GrabCut database that we used for the parameter sensitivity study. Figure 6.5 (a) has very well defined boundaries where as the leopard in figure 6.5 (c) camouflages into the background. From figure 6.6 we see that the parameter σ does not have much effect on the segmentation when boundaries are smooth and well defined. For less well defined boundaries it has a more pronounced effect. Smaller values of σ can produce less smooth boundaries. This is because the Gaussian potentials defining the pixel-wise similarities is more heavily peaked. This allows better discrimination between pixels hence fine details in the image are better captured.

6.5 Discussion

In this chapter we have shown that the mechanisms of the PageRank denoising and Power Iteration denoising algorithms can be used in an interactive image segmentation framework. We showed explicitly how the PageRank denoising algorithm relates to the semi-supervised learning algorithm of [Zhu *et al.* (2003a)] and the random walker segmentation algorithm of [Grady (2006)]. We have seen that the PageRank denoising algorithm outperforms the method proposed in [Grady (2006)] and is competitive against graph cut methods. Moreover we believe the performance of the algorithm can be improved by boosting the discriminative

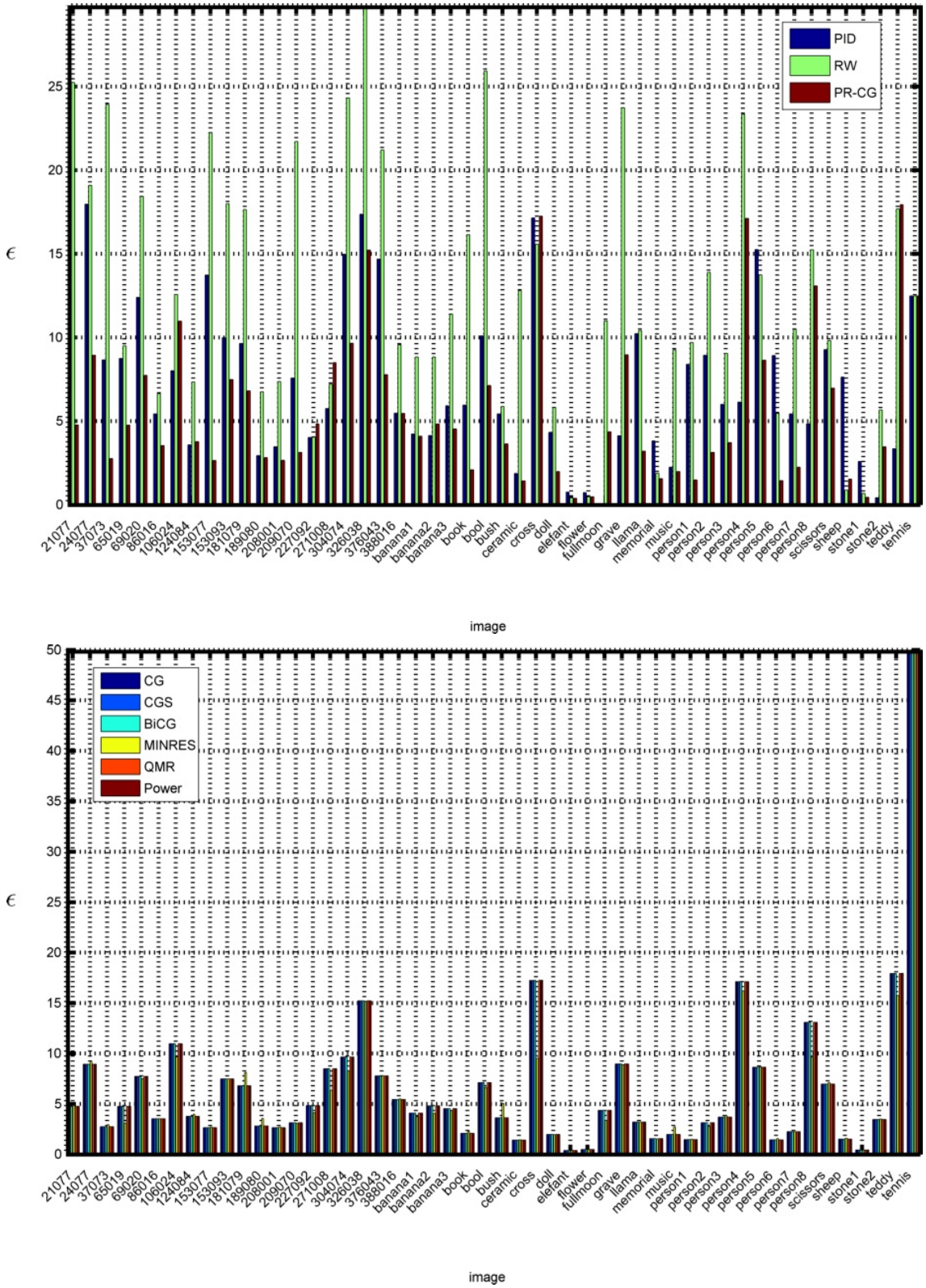


Figure 6.3: Misclassification error rates as a percentage.

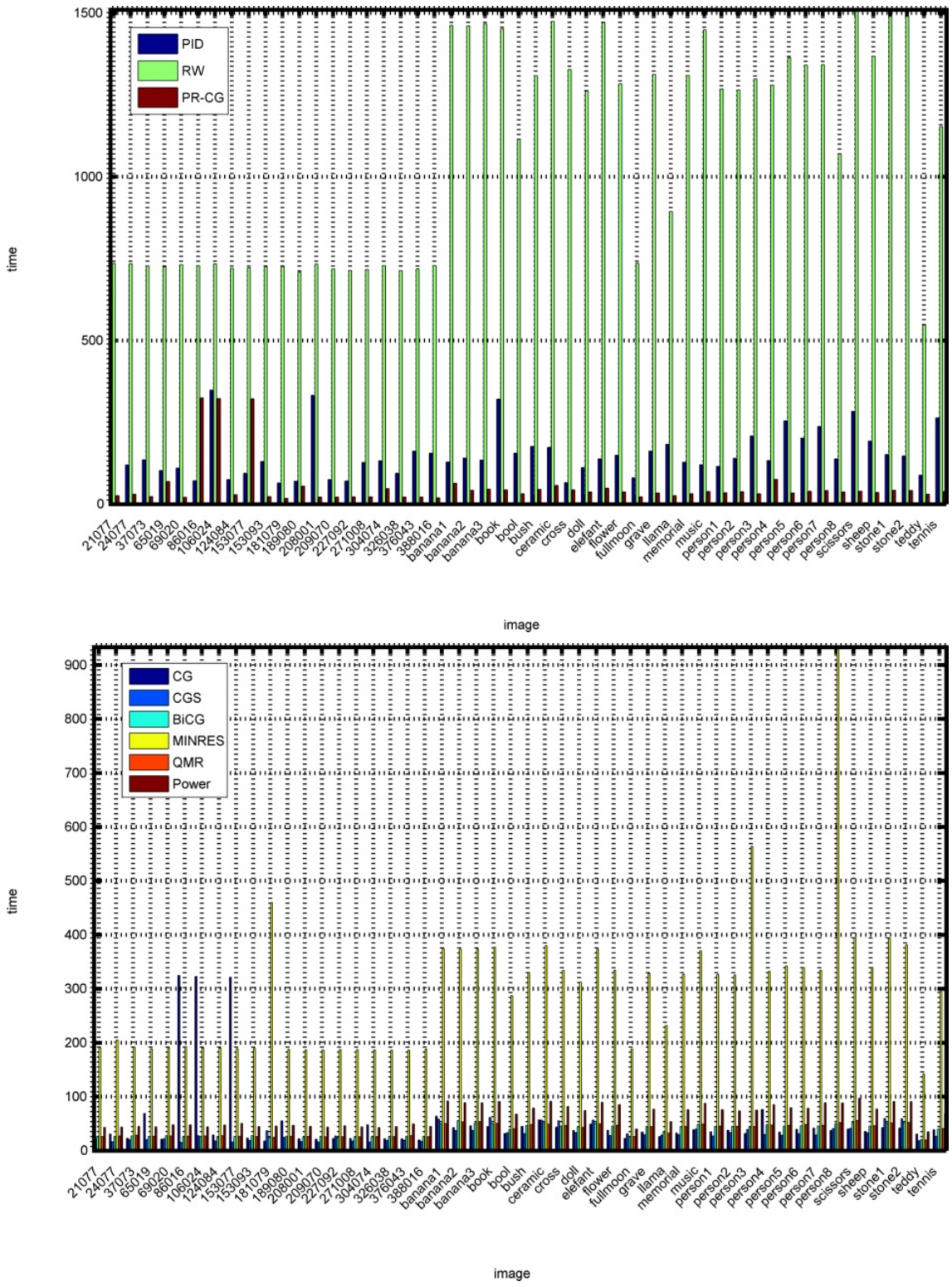


Figure 6.4: Run times in seonds.

Method	Average misclassification %	Median misclassification %
RW	12.34	10.46
PID	7.26	5.99
PR-CG	5.48	3.95
PR-CGS	5.48	3.95
PR-BiCG	5.48	3.95
PR-MINRES	5.10	3.86
PR-QMR	5.48	3.95
PR-Power	5.48	3.95

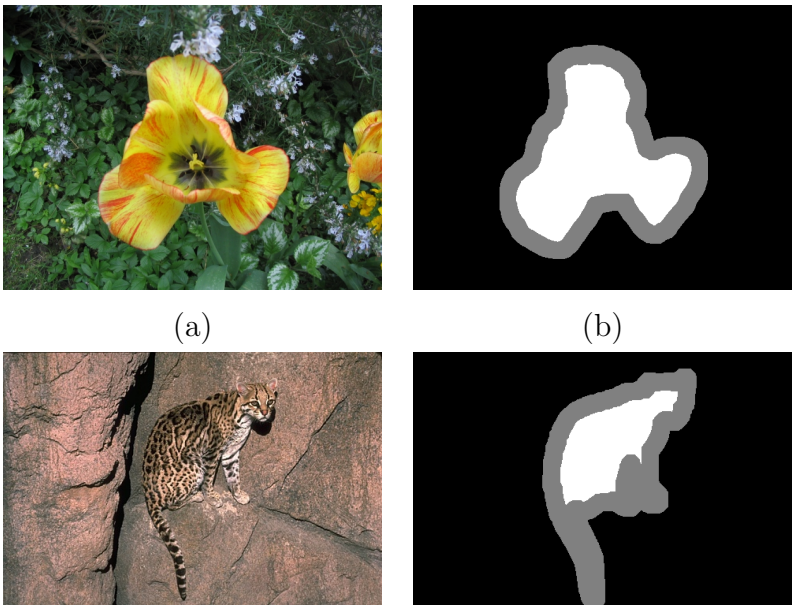
Table 6.1: Average and median misclassification rates evaluated over the Microsoft GrabCut database. The algorithms are random walker (RW) Power Iteration denoising (PID), PageRank denoising (PR) and PageRank denoising variants.

Method	Average run time (s)	Median run time (s)
RW	1612	1135
PID	150.10	135.55
PR-CG	53.84	37.55
PR-CGS	31.30	29.28
PR-BiCG	37.81	42.23
PR-MINRES	296.22	319.19
PR-QMR	38.27	42.12
PR-Power	65.84	71.79

Table 6.2: Average and median timings evaluated over the Microsoft GrabCut database. The algorithms are random walker (RW) Power Iteration denoising (PID), PageRank denoising (PR) and PageRank denoising variants.

Method	Average misclassification %
GMMRF	7.20
GrabCut	7.90
PID	7.26
PR	5.48

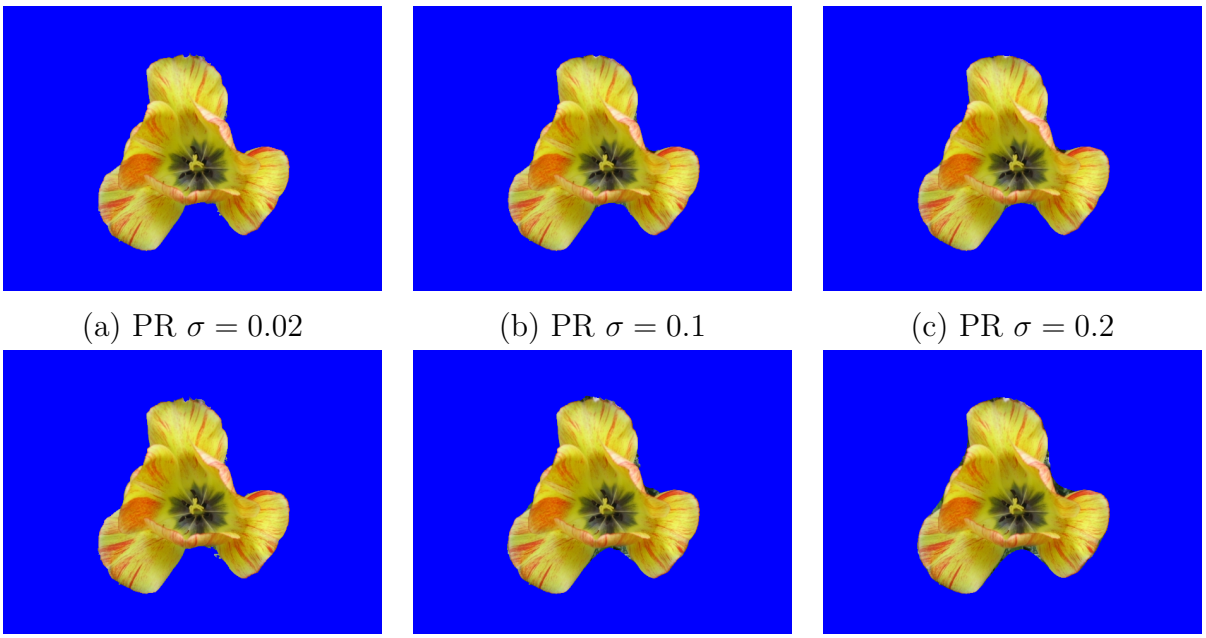
Table 6.3: Average misclassification rates evaluated over the Microsoft GrabCut database versus graph cut methods: GMMRF [Blake *et al.* (2004)] GrabCut [Rother *et al.* (2004)].



(a) (b)

(c) (d)

Figure 6.5: Images and trimaps from Microsoft GrabCut database used for study of the parameter sensitivity of the PageRank denoising and Power Iteration denoising algorithms applied to image segmentation. (a) Flower and (b) its associated trimap. (d) 326038 and its associated trimap.



(a) PR $\sigma = 0.02$ (b) PR $\sigma = 0.1$ (c) PR $\sigma = 0.2$

(d) PID $\sigma = 0.02$ (e) PID $\sigma = 0.1$ (f) PID $\sigma = 0.2$

Figure 6.6: Parameter sensitivity study on flower figure from the Microsoft GrabCut database. (a) $\epsilon = 0.49\%$, (b) $\epsilon = 0.56\%$ and (c) $\epsilon = 0.68\%$. (c) $\epsilon = 0.75\%$, (b) $\epsilon = 3.23\%$ and (c) $\epsilon = 5.28\%$ where ϵ is the segmentation error rate defined in equation 6.24.

(a) PR $\sigma = 0.02$ (b) PR $\sigma = 0.1$ (c) PR $\sigma = 0.2$

(d) PID $\sigma = 0.02$ (e) PID $\sigma = 0.1$ (f) PID $\sigma = 0.2$

Figure 6.7: Parameter sensitivity study on 326038 figure from the Microsoft GrabCut database. (a) $\epsilon = 15.23\%$, (b) $\epsilon = 14.11\%$ and (c) $\epsilon = 14.32\%$. (c) $\epsilon = 17.37\%$, (b) $\epsilon = 17.44\%$ and (c) $\epsilon = 17.289\%$ where ϵ is the segmentation error rate defined in equation 6.24.

performance of the classifier.

Although we have performed some experiments on the parameter sensitivity of our proposed algorithms it is to be noted that an evaluation on user interaction required to produce "good" segmentation could greatly improve our understanding of the utility of the algorithms. Unfortunately, to the best of our knowledge, a useful measure does not exist in the literature. Moreover a database with relevant brush-strokes has not been devised. Useful efforts for future work would be to develop such a measure and database. This would significantly enhance the literature and provide a good basis for scientific study of related methods.

Significance. This chapter is significant in this thesis as it provides experimental evidence to support the main thesis claim that

> *Graph based diffusion processes provide a rich detail preserving scale-space for the development of image based pattern recognition and manipulation algorithms.*

Specifically it shows how the graph based methods can be used to effect images based pattern recognition tasks such as segmentation. We believe we have provided sufficient evidence to support this claim. Interestingly we noted that with very minor or no modification to the PageRank denoising framework we were able to effect both denoising and segmentation. This property is shared with partial differential equation based diffusion processes used in active contour models. Therefore before we get into the conclusions of this thesis let us discuss some relationships with the active contour methods.

Relation to active contour methods. Active contour methods are energy-based methods that seek to

> *find groups of pixels that "go together"*

by minimising an energy functional

$$E = E_{data} + E_{smooth} \tag{6.25}$$

where the data term enforces that the pixels are similar to the data and the smoothness term imposes constraints on the similarity of the pixels in a region or boundary. In the modern literature [Chan & Vese (2001)] the data term is

based upon region statistics interior and exterior to the region of interest. If we denote a foreground object of interest as bounded by a contour C with the interior region denoted as C_1 and the exterior as C_2 then the data term seeks to ensure that pixels in region C_1 share similar statistics, for example average intensity, and pixels in region C_2 share similar statistics. Optimisation of these methods is carried out using a diffusion process simulated with the level set method.
In our probabilistic interactive methods our brush-strokes denote training data that is used to "learn" the region statistics of a user defined regions C_1 and C_2. Based upon the connectivity of the graph a diffusion process insures the grouped pixels have similar statistical properties.

Chapter 7

Conclusions

This thesis has presented arguments, experiments and evidence supporting the claim that

> *Graph based diffusion processes provide a rich detail preserving scale-space for the development of image based pattern recognition and manipulation algorithms.*

To validate this claim we developed novel edge-preserving smoothing algorithms based around the random walk graph Laplacian. The algorithms were then applied in areas of computational photography that require multi-scale image decompositions for tone management and mapping. Finally we showed that these edge-preserving filter algorithms can be used as diffusion processes in an interactive (semi-supervised) image segmentation framework. The methods are therefore flexible and have interesting implications for future work. Moreover we note that even though we used diffusion processes analogous to partial differential equation methods we did not have to discretise the operators avoiding the stability and boundary conditions imposed by discretisation of the partial differential equation operators.

Specifically evidence and support for the central claim of this thesis is provided by the key contributions

1. the development of two novel edge-preserving filter algorithms

2. application of the edge-preserving filter algorithms to problems in computational photography

3. extension of the associated diffusion processes to interactive (semi-supervised) image segmentation.

7.0.1 Evidence and Support

The claim that

> *Ggraph based diffusion processes provide a rich detail preserving scale-space for the development of image based pattern recognition and manipulation algorithms.*

can be divided and analysed as two statements:

(1) **Graph based diffusion processes provide a rich detail preserving scale-space.**

This statement can be used to provide *support* for our main thesis claim. This claim draws support from chapters 2 and 4. Firstly in chapter 2 we undertook a survey of the literature on mathematical models for low-level vision with emphasis on diffusion processes and graph based models. We understood, from the literature, that the goal of such methods is to

> *smooth noisy variations in the image whilst preserving meaningful structure.*

This paradigm is often, more succinctly, referred to in the literature as *edge-preserving filtering* or *edge-preserving smoothing*. We also considered, in chapter 2, connections between graph based diffusion processes and partial differential equation processes through the notion of the Laplacian.
In chapters 4 and 5 we considered frameworks for deriving graph based diffusion processes. We motivated arguments based on Markov random fields as such energy-based minimisation methods are de facto in computer vision. We the proposed and derived two new graph based edge-preserving filters which we named

PageRank denoising (PR) and Power Iteration denoising (PID). To relate these methods to scale-space theories and give intuitive arguments as to the behaviour of the models we used an electrical network analogy. The electrical network analogy is based upon arguments from Ohm's law and Kirchoff's current laws. This is not arbitrary as such arguments were used in the scale-space literature to describe the behaviour of partial differential equation based scale-space methods. Further to this these analogies provide good theoretical links to graph based methods in machine learning. Finally these edge-preserving filters can be seen to be supported by the following axioms for linear scale-space representations

1. **Causality**. A scale-space representation should have the property that no spurious detail should be generated passing from finer to coarser scales.

2. **Immediate Localisation.** At each resolution, region boundaries should be sharp and coincide with the semantically meaningful boundaries at that resolution

3. **Piecewise smoothing.** At all scales intra-region smoothing should occur preferentially over inter-region smoothing.

(2) **The development of image-based pattern recognition and manipulation algorithms.**

We use this statement to investigate the experimental evidence in this thesis to support the main thesis claim. In particular chapter 4 provides extensive experimental validation of the proposed edge-preserving filter algorithms. We discover the limitations of these algorithms with respect to the denoising problem but discover their utility in providing multi-scale decompositions. We find that these algorithms are well suited for computational photography applications where detail preserving multi-scale image decompositions are required. Particularly in the areas of image tone manipulation and tone mapping of high dynamic range images. We show that our methods are competitive with respect to the de facto Bilateral filter and sate-of-the-art methods based upon the weighted least squares method. Our methods are easier to control as they only require the specification of one parameter. This provides evidence that the rich scale-spaces allow

us to manipulate aspects of the image. Chapter 5 uses the diffusion process in the pattern recognition task of image segmentation. We find that our framework is directly suitable for a semi-supervised (interactive) image segmentation task with interesting implications on future work.

7.0.2 Future Work

In this thesis we showed that Graph based diffusion processes can provide a rich detail preserving scale space for the development of image based pattern recognition and manipulation algorithms. Moreover these methods exhibit application performance levels comparable with de facto and state-of-the-art methods in a variety of applications whilst remaining generic and flexible. Nevertheless the presented approach has a number of limitations which motivate further research in this area. We identify four key areas under the headings of computational speed, graph construction, interactive performance and boosting that may provide interesting and fruitful directions for further research and investigation.

Computational speed. Firstly one may wish to address the computational speed of the algorithms. We provided computationally efficient algorithms based on computational linear algebra to affect the solution of the associated linear systems and presented results as to determine the most efficient Krylov subspace solver. Our methods did not take advantage of general purpose graphics processor unit (GPGPU) implementations. Recent studies [Buatois *et al.* (2009)] have shown that Krylov subspace solvers can efficiently be implemented on GPGPUs greatly speeding up performance. They show that speeding up the sparse matrix-vector multiplication (SpMV) at the core of the linear solvers is the key to speeding up these iterative solvers. Speeding up this operation requires one to use data structures that fit the highly parallel architecture of the modern GPGPU. At the heart of the algorithms, we developed, is the update

$$\mathbf{v}^{n+1} \leftarrow M_{rw}\mathbf{u}^n \tag{7.1}$$

where M_{rw} is the Nadaraya-Watson kernel estimator. This is the SpMV operation that is the computational bottleneck of our algorithms. To speed up the algorithm we require that we speed up this multiplication. We notice that this

update is present in the non-local means algorithm and the Bilateral filter methods. There has been considerable recent interest and work dedicated to speeding up this operation for the Bilateral filter and non-local means algorithms [Adams *et al.* (2010),Adams *et al.* (2009),Paris & Durand (2009)] where interactive speeds were achieved for colour images on GPGPU hardware. We can therefore argue that employing their proposed update strategies could conceivably achieve real-time/interactive speeds for our algorithms. The active learning framework allows the machine learning algorithm to pick a set of unlabeled instances to be labeled by the domain expert (artist) which are then used to augment the initially supplied training data (brush-strokes). There has been a great deal of research in this area of machine learning with [Zhu *et al.* (2003b)] applying these concepts to the semi-supervised machine learning problem. In brief their framework allows one to efficiently estimate the expected generalisation error after querying a point. Once the queries are selected and added to the labeled data the classifier can then be trained using both the labeled training data (brush-strokes) and the remained unlabeled pixels. This strategy may have an impact on problems where the classifier may struggle to classify certain regions; in which case the active learning framework may allow the algorithm to query the oracle (artist) for further brush-stroke inputs.

Graph Construction. In this thesis we have not given much attention to the graph construction problem. Particularly we defined the weights between nodes using Gaussian potential functions of the form

$$w_{ij} = \exp\left[-\frac{(u_i - u_j)^2}{\sigma^2}\right] \tag{7.2}$$

where σ is the scale parameter. This is a fairly standard construction in computer vision and image processing problems [Shi & Malik (2000)]. We then used σ as a parameter to control the scale-spaces of the edge-preserving smoothers. We have seen that the choice of σ has a great impact on the output of the edge-preserving smoothing algorithms. An interesting direction for further investigation would be to allow a machine learning algorithm such as dimensionality reduction methods: local linear embedding (LLE) [Rowels & Saul (2000)] and ISOMAP [Tenenbaum *et al.* (2000)], to learn a graph embedding. For example using the local linear

embedding algorithm (LLE) we could define a reconstruction error as

$$\epsilon = \sum_i \|f_i - \sum_j w_{ij} f_j\|_2^2 \tag{7.3}$$

where the vector $\mathbf{f} = (f_1, f_2, \ldots, f_N)$ is the image data. We then seek the weights w_{ij} that minimise this error subject to the constraints

$$\begin{aligned} \textstyle\sum_j w_{ij} &= 1 \\ w_{ij} &\geq 0. \end{aligned} \tag{7.4}$$

This quadratic programme can be solved efficiently with one of many standard optimisation algorithms such as the active set method [Nocedal & Wright (1999)]. This is not the only machine learning strategy that we could employ to construct the graph embedding. There has been recent work and interest in this graph construction problem [Yan *et al.* (2007),Yan & Wang (2009)] which may have impact on the methods developed in this thesis.

Interactive performance. Interactive speeds would greatly enhance the applicability of our methods. For example an interesting direction would be that of developing interactive local tonal adjustment algorithms. Such tools for effecting local tonal adjustments allow for creative expression or "perfection" of digital photography. These tools are valuable in that they allow the photographer or artist to quickly indicate regions of interest by drawing a few single brush strokes and allowing an edge-preserving smoother to locally propagate the tonal adjustments, much like in our developed semi-supervised image segmentation algorithm. This is opposed to the viewpoint of carefully selecting regions of interest by drawing or hand-painting layer masks. In essence the purpose of such tools is to enable the artist to easily adjust exposure or other effects to reveal details, enhance contrast or modify colour saturation with minimal user interaction. A framework could have the workflow

1. load the digital image or HDR radiance map
2. using brush strokes indicate regions requiring editing
3. experiment with adjustment values and parameters until a satisfactory result is achieved

4. iterate steps 2 and 3 until the desired effects are achieved,

as proposed in [Lischinski *et al.* (2006)]. In their approach to affect the spatially varying exposure function f, which is a scalar that specifies (in terms of f-stops) how the exposure of each pixel is to be adjusted, they propose to achieve this goal by minimizing the following functional for the argument f

$$\lambda \sum_{ij} h(f, L) + \sum_{i} c_i (f_i - u_i) \tag{7.5}$$

where c_i is a soft constraint (between 0 and 1) indicating which pixels are constrained by the brush strokes. Hence the right hand term is a data fidelity term. The left hand term controls the "diffusion" or propagation of the constrained exposure information according to gradients in the log-luminance channel. We note that we could implement such a function in our PageRank denoising framework giving

$$\sum_{ij} \frac{w_{ij}}{\sum_{ij} w_{ij}} (u_i - u_j) + \sum_{ij} c_i (f_i - u_i). \tag{7.6}$$

Which can be solved using the PageRank denoising algorithm. The right hand term is the data fidelity term with c_i (between 0 and 1) as the soft constraint and the left hand term is the smoothing term that generates the edge-preserving smoothing effect where the weights w_{ij} can be generated from the gradients in the log-luminance channel. The advantage of our framework when applied to such problems and indeed the interactive image segmentation problem is that we could appeal to the active learning framework for machine learning.

Boosting. Finally at a high level the PageRank denoising and Power Iteration denoising algorithms work by propagation of information using the connectivity structure of the graph Laplacian. This allows the algorithms to produce the edge-preserving smoothing effect. A problem with this approach and in general related approaches is that texture edges within object boundaries may slow propagation. If these texture edges are weak as compared to relative object boundaries then this may not be a problem. Another problem is that the manipulation of fragmented or occluded appearance, such as, for example, blue sky peeking through the leaves of a tree, may be challenging particularly for the semi-supervised (interactive) image segmentation problem since the influence of scribbled brush-strokes may be

stopped by edges in-between. To overcome this a user might have to scribble in-between edges which may be time consuming or cumbersome. To overcome and improve the performance of our algorithm in such cases it may be useful to use the machine learning concept of boosting. Boosting operates on a principle that "good" classifiers can be built as a weighted combination of many "weak" classifiers each of which performs just better than random guesses on the training data. The brush-strokes (or training data) can be used to train a boosting algorithm to enhance discrimination of image regions. Such ideas have recently been used in a similar framework known as ScribbleBoost [Li *et al.* (2008)] based upon the GentleBoost algorithm [Friedman *et al.* (2000)]. Boosting has also been used in other areas of computer vision such as object detection and classification [Leistner *et al.* (2008)] and clustering in image retrieval [Hertz *et al.* (2004)].

7.1 Closing remarks

In the beginning we considered the wisdom of the Ecclesiastic philosophers who spoke the words

> "*In searching many books there is no end*".

Indeed in studying the subject of *image processing* processing we have traversed many esoteric subjects in mathematical sciences from high energy physics to computational linear algebra to Markov processes; mentioning just a few. The study was only in the context of a mathematical framework; we did not even touch the surface of vision science and philosophical aspects. The modern subject is extensive and vast with niche contributions that may require the novice to gain a level of expertise and skill in challenging subjects. It is in the *inspired* contributions of some authors that lead us to consider the subject in wider contexts and link to other areas of science. Two contributions, in the literature, I would like to high light are the *high energy physics* formulation of the image flows and scale spaces and the relation of data to *discrete manifolds embedded in higher dimensional Riemannian manifolds*. These contributions invoke the memory of the words of William Blake

"*Improvements make straight roads; but the crooked roads without improvements are roads of Genius*"

and are the contributions from the literature that inspire my work.
In general there is no one dominant framework or model or framework. The subject is active, fluid and continuously evolving. We conclude this thesis with the words of the statistician George Box

"*Essentially, all models are wrong, but some are useful*".

References

Adams, A., Gelfand, N., Dolson, J. & Levoy, M. (2009). Gaussian kd-trees for fast high-dimensional filtering. In *SIGGRAPH*. 170

Adams, A., Baek, J. & Davis, M.A. (2010). Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, **29**, 753 – 762. 170

Alexander, V. & Kaleem, S. (2002). Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 1565 – 1578. 41

Alvarez, L. & Morel, J.M. (1994a). *Morphological approach to multiscale analysis: From principles to equations, In Geometric-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, The Netherlands. 1, 18

Alvarez, L. & Morel, J.M. (1994b). Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, **31**, 590 – 605. 136

Aujol, J.F. (2009). Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, **34**, 307–327. 14, 35, 66

Azzabou, N., Paragios, N. & Guichard, F. (2006). Random walks, constrained multiple hypothesis testing and image enhancement. In *9th European Conference on Computer Vision*. 67

Bae, S., Paris, S. & Durand, F. (2006). Two-scale tone management for photographic look. *ACM Transactions on Graphics*, **25**, 654 – 662. 128, 130, 145

Bai, X. & Sapiro, G. (2007). A geodesic framework for fast interactive image and video segmentation and matting. In *International conference on computer vision*. 153

Ballerini, L. (2001). Genetic snakes for color images segmentation. *Applications of Evolutionary Computing, Lecture notes in computer science*, **2037/2001**, 268 – 277. 41

Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D. & Aly, M. (2008). Video suggestion and discovery for youtube: Taking random walks through the view graph. In *WWW-2008*. 90

Barrett, B., Berry, M., Chan, T.F., Demmel, J., Donato, J.M., Dongarra, J., Eijkhout, V., R, P., Romine, C. & Van der Vorst, H. (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.*. SIAM. xiv, 108, 109

Belkin, M. & Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing System*, **14**, 585 – 591. 5, 15

Benzi, M., Golub, G.H. & Liesen, J. (2005). Numerical solution of saddle point problems. *Acta Numerica*, **14**, 1–137. 108

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, **48**, 259 – 302. 15, 35

Bishop, C. (2007). *Pattern Recognition and Machine Learning*. Springer. 15, 27, 28, 83

Blake, A., Rother, C., Brown, M., Perez, P. & Torr, P. (2004). Interactiv image segmentation using an adaptive gmmrf model. In *European Conference on Computer Vision*. xv, 43, 153, 155, 157, 161

Blum, A. & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Eighteenth International Conference on Machine Learning*. 154

Blum, A., Lafferty, M.R., J. Rwebangira & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In *International conference on Machine learning*. 153, 154

Bougleux, S., Elmoataz, A. & Melkemi, M. (2009). Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *International Journal of Computer Vision*, **84**, 220 – 236. 35, 67, 71

Boykov, Y. & Funka-Lea, G. (2006). Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, **70**, 109 – 131. 154

Boykov, Y. & Jolly, M.P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE International Conference on In Computer Vision*. 16, 43, 46, 64, 153, 154

Boykov, Y. & Komolgorv, V. (2004). An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1124 – 1137. 46, 154

Bresson, X. & F, C.T. (2008). Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging*, **2**, 455–484. 66

Bresson, X., Vandergheynst, P. & Thiran, J. (2006). Multiscale active contours. *International Journal of Computer Vision*, **70**, 197 – 211. 3

Brice, C.R. & Fennema, C.L. (1970). Scene analysis using regions. In *Artificial Intelligence*. 39

Briggs, W.L., Henson, V.E. & McCormick, S.F. (2000). *A multigrid tutorial*. SIAM. 45, 59

Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual web search engin. *Computer Networks and ISDN Systems*, **30**, 107 – 117. 8, 23, 65, 69, 70, 90

Buades, A., Coll, B. & Morel, J.M. (2005a). A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 27, 36, 66, 67, 73, 80, 85, 96, 115, 117

Buades, A., Coll, B. & Morel, J.M. (2005b). A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, **4**, 490 – 530. 67

Buatois, L., Caumon, G. & Levy, B. (2009). Concurrent number cruncher - a gpu implementation of a general sparse linear solver. *International Journal of Parallel, Emergent and Distributed Systems*, **24**, 205 – 223. 169

Carreira-Perpinan, A., M (2006). Fast nonparametric clustering with gaussian blurring mean-shift. In *International Conference on Machine Learning*. 83

Caselles, V., Kimmel, R. & Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, **22**, 61– – 79. 40, 41

Chambolle, A. (2004a). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, **20**, 89–97. 14, 66, 125

Chambolle, A. (2004b). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, **20**, 89–97. 66, 115

Chan, R. & Jin, X. (2007). *An Introduction to Iterative Toeplitz Solvers*. SIAM. 51, 53, 54, 108

Chan, R. & Ng, M.K. (1996). Conjugate gradient methods for toeplitz systems. *SIAM Review*, **38**, 427–448. 53, 54

Chan, T.F. & Vese, L.A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, **10**, 266 – 277. 14, 43, 64, 164

Chan, T.F., Shen, J. & Vese, L. (2003). Variational pde models in image processing. *Notices of the American Mathematical Society*, **50**, 14–26. 14

Chatterjee, P. & Milanfar, P. (2010). Is denoising dead. *IEEE Trans. on Image Processing*, **19**, 895 – 911. 36, 37, 125

Chen, S.S., Donoho, D.L. & Saunders, M.A. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, **43**, 129 – 159. 32

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 790 – 799. 27, 83

Chung, F. (1997). *Spectral Graph Theory*. American Mathematical Society. 5, 15, 35, 67, 74, 79, 80

Chung, F. (2007). The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, **104**, 19735–19740. 71

Cohen, L. (1991). On active contour models and balloons. *CVGIP Image Understanding*, **53**, 211–218. 41

Cohen, L. & Cohen, I. (1993). Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1131–1147. 41

Cohen, W.W. & Lin, F. (2010). Power iteration clustering. In *International Conference on Machine Learning*. 96, 97

Coifman, R.R. & Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, **21**, 5 – 30. 15, 70, 80

Comaniciu, D. & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 603 619. 39, 83, 134

Couprie, C., Grady, Najman, L. & Talbot, H. (2009). Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *IEEE International Conference on Computer Vision*. 6, 47

Cremers, D., Rousson, M. & Deriche, R. (2007). A review of statistical approaches to level set segmentation : Integrating color, texture, motion and shape. *International journal of computer vision*, **72**, 195 – 215. 39, 43, 64

Cui, J., Yang, Q., Wen, F., Wu, Q., Zhang, C., Van Gool, L. & Tang, X. (2008). Transductive object cutout. In *IEEE Conference on Computer Vision and Pattern Recognition*. 46

Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, **16**, 2080 – 2095. 37

Debevec, P. & Malik, J. (1997). Recovering high dynamic range radiace maps from photographs. In *SIGGRAPH*. 128, 129

DeCarlo, D. & Santella, A. (2002). Stylization and abstraction of photographs. *ACM Transations on Graphics*, **21**, 769 – 776. 118, 120, 132

Donoho, D.L. (1995). Denoising by soft thresholding. *IEEE Transactions on Information Theory*, **41**, 613 – 627. 36

Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J. & Segonne, F. (2008). Segmentation by transduction. In *IEEE Conference on Computer Vision and Pattern Recognition*. 80, 153

Durand, F. & Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics*, **27**, 257 – 266. 128, 129, 132, 145

Eisemann, E. & Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics*, **23**, 673 – 678. 128, 129, 142

Elad, M. & Aharon, M. (2006). Image denoising via learned dictionaries and sparse representation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 33, 37

Elmoataz, A., Lezoray, O. & Bougleux, S. (2008). Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Transactions on Image Processing*, **17**, 1047–1060. 35, 67, 68, 71, 72, 99, 100, 115, 125

Esedoglu, S. & Shen, J. (2002). Digital inpainting based on the mumford-shah-euler image model. *European Journal of Applied Mathematics*, **13**, 353 – 370. 14

Estrada, D., F Fleet & Jepson, A. (2009). Stochastic image denoising. In *British Machine Vision Conference*. 67

Farbman, Z., Fattal, R., Lischniski, D. & Szeliski, R. (2008). Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM SIGGRAPH*. xi, xii, 129, 130, 132, 133, 134, 135, 136, 138, 139, 140

Fattal, R., Agrawala, M. & Rusinkiewicz, S. (2007). Multiscale shape and detail enhancement from multi-light image collections. In *ACM SIGGRAPH*. 128, 134, 145

Felzenszwalb, P.F. & Huttenlocher, D.P. (2004). Efficient belief propagation for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition*. 16, 78

Fowlkes, C., Belongie, S., Chung, F. & Malik, J. (2004). Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 214 – 225. 15, 45

Friedman, J., T, H. & Tibshirani, R. (2000). Additive logistic regression, a statistical view of boosting. *Annals of Statistics*, **28**, 337 – 374. 173

Fukunaga, K. & Hostetler, L.D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, **21**, 32 – 40. 27, 82, 83

Geman, S. & Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**, 6. 35

Gil, D. & Radeva, P. (2003). Curvature vector flow to assure convergent deformable models for shape modelling. *Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science*, **2683/2003**, 357 – 372. 41

Gilboa, G. & Osher, S. (2007). Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, **7**, 1005 – 1028. 67

Golub, G.H. & Van Loan, C.F. (1996). *Matrix computations*. Johns Hopkins Studies In The Mathematical Sciences. 66, 108

Gomo, P. (2010). Pagerank image denoising. *Lecture Notes in Computer Science 6111*, 1 – 10. 8

Gomo, P. & Spann, M. (2009). Fft-based conjugate gradient strategies for numerical simulation of gvf-snakes. In *1st International Conference on Computational and Mathematical Biomedical Engineering*. 8

Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 1768 – 1783. 6, 15, 47, 148, 153, 154, 155, 157

Grady, L. & Alvino, C. (2009). The piecewise smooth mumford-shah functional on an arbitrary graph. *IEEE Trans. on Image Processing*, **In press**. 6, 15, 47

Grady, L. & Schwartz, E.L. (2006). Isoperimetric graph partitioning for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 469 – 475. 6, 15

Grady, L., Sun, Y. & Williams, J. (2006). *Three Interactive Graph-Based Segmentation Methods Applied to Cardiovascular Imaging In Mathematical Models in Computer Vision: The Handbook*. Springer. 148

Greig, D.M., Porteous, B.T. & Seheult, A.H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, **51**, 271 – 279. 16, 46, 64

Haber, E. & Modersitzki, J. (2006). A multilevel method for image registration. *SIAM Journal on Scientific Computing*, **27**, 1594–1607. 42

Han, X., Xu, C. & Prince, J. (2007). Fast numerical scheme for gradient vector flow computation using a multigrid method. *IET Image Processing*, **1**, 48–55. 42, 49, 56, 59, 60

Hassouna, M. & Farag, A. (2009). Variational curve skeletons using gradient vector flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**, 2257 – 2274. 48

Haveliwala, T. & Kamvar, S. (2003). The second eigenvalue of the google matrix. Tech. rep., Stanford. 83, 108

Heers, J., Schnörr, C. & Stiehl, H.S. (2001). Globally convergent iterative numerical schemes for nonlinear variational image smoothing and segmentation on a multiprocessor machine. *IEEE Transactions on Image Processing*, **10**, 852–864. 42

Hein, M., Audibert, J.Y. & Von Luxburg, U. (2005). From graphs to manifolds weak and strong pointwise consistency of graph laplacians. *Lecture notes in computer science*, **3559**, 470 – 485. 15, 69, 70, 71

Henn, K., S.and Witsch (2001). Iterative multigrid regularization techniques for image matching. *SIAM Journal on Scientific Computing*, **23**, 1077–1093. 42

Hertz, T., Bar-Hillel, A. & Weinshall, D. (2004). Learning distance functions for image retrieval. In *Computer Vision and Pattern Recognition*. 173

Hestenes, M.R. & Stiefel, S. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, **49**, 409–436. 66

Horn, B.K.P. & Schunck, B.G. (1982). Determining optical flow. *Shape recovery, Physics-Based Vision: Principles And Practice*, 389 – 407. 14

Horowitz, S.L. & Pavlidis, T. (1976). Picture segmentation by a tree traversal algorithm. *Journal of the ACM*, **23**, 368 – 388. 39

Jain, A.K., Topchy, A., Law, M.H.C. & Buhmann, J.M. (2004). Landscape of clustering algorithms. In *International Conference on Pattern Recognition*. 39

Jeh, G. & Widom, J. (2003). Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*. 84

Juan, O., Keriven, R. & Postelnicu, G. (2003). Stochastic mean curvature motion in computer vision: Stochastic active contours. In *2nd IEEE Workshop on Variational, Geometric & Level Set Methods*. 41

Juan, O., Keriven, R. & Postelnicu, G. (2006). Stochastic motion and the level set method in computer vision: Stochastic active contours. *International Journal of Computer Vision*, **69**, 7 – 25. 41

Kang, S.B., Uyttendaele, M., Winder, S. & Szeliski, R. (2003). High dynamic range video. In *SIGGRAPH*. 129

Kass, M., Witkin, A. & Terzopoulos., D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, **1**, 321–331. 39, 41

Köstler, H., Ruhnau, K. & Wienands, R. (2008). Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer. *Numerical Linear Algebra with Applications*, **15**, 201–218. 42

Khan, E.A., Reinhard, E.F. & Fleming, R.W. (2006). Image-based material editing. *ACM Transactions on Graphics*, **25**, 654 – 663. 128

Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A. & Yezzi, A. (1995). Gradient flows and geometric active contour models. In *Fifth International Conference on Computer Vision, 1995. Proceedings*. 40, 41

Kimmel, R., Malladi, R. & Sochen, N. (1997). Images as embedding maps and minimal surfaces: Movies, color, and volumetric medical images. In *IEEE Computer Vision and Pattern Recognition*. 4, 21

Kimmel, R., Malladi, R. & Sochen, N. (2000). Images as embedded maps and minimal surfaces: Movies, color, and volumetric medical images. *International Journal of Computer Vision*, **39**, 111 – 19. 4, 21

Kindermann, S., Osher, S. & Jones, P.W. (2005). Deblurring and denoising of images by nonlocal functionals. *SIAM: Multiscale Modeling and Simulation*, **4**, 1091 – 1115. 15

Koenderink, J.J. (1984). The structure of images. *Biological Cybernetics*, **50**, 363 – 370. 18

Kohlberger, T., Schnörr, C., Bruhn, A. & Weickert, J. (2005). Domain decomposition for variational optical-flow computation. *IEEE Transactions on Image Processing*, **14**, 1125–1137. 42

Komolgorv, V. & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 147 – 159. 16, 78

Kondor, R. & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. In *ICML*. 69

Kumar, S. & Hebert, M. (2006). Discriminative random fields. *International Journal of Computer Visi*, **68**, 179 – 201. 95

Kumar, S. & Martial, H. (2003). Discriminative fields for modeling spatial dependencies in natural images. In *In Advances in Neural Information Processing Systems*. 36, 95

Lafferty, J. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on on Machine Learning*. 36, 95

Lee, S.J. (1983). Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, **24**, 255 – 269. 27

Leistner, C., Garbner, H. & Bischof, H. (2008). Semi-supervised boosting using visual similarity learning. In *Computer Vision and Pattern Recognition*. 173

Lempel, R. & Moran, S. (2000). The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, **33**, 387 – –401. 70

Lempitsky, V.S. & Boykov, Y. (2007). Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*. 46

Lempitsky, V.S., Blake, A. & Rother, C. (2008). Image segmentation by branch-and-mincu. In *European Conference on Computer VIsion*. 46

Leroy, B., Herlin, I.L. & Cohen, L.D. (1996). Multi-resolution algorithms for active contour models. *Lecture Notes in Control and Information Sciences*, **Volume 219/1996**, 58 – 65. 41

Levin, A., Lischinski, D. & Weiss, Y. (2008). A closed form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Interlligence*, **30**, 228 – 242. 148

Levoy, M. (2010). Experimental platforms for computational photography. *IEEE Computer Graphics and Applications*, **30**, 81 – 87. 128, 142

Li, B. & Acton, S.T. (2007). Active contour external force using vector field convolution for image segmentation. *IEEE Trans. on Image Processing*, **16**, 2096 – 2106. 41

Li, Y., Adelson, E.H. & Agarwa, A. (2008). Scribbleboost: Adding classification to edge-aware interpolation of local image and video adjustments. *Computer Graphics Forum*, **27**, 1255–1264. 173

Lin, F.R. (2001). Preconditioners for block toeplitz systems based on circulant preconditioners. *Numerical Algorithms*, **26**, 365–379. 53, 54

Lindenbaum, M. & Fischer, A., M andBruckstein (1994). On gabor's contribution to image enhancement. *Pattern Recognition*, **21**, 1 – 8. 18, 19

Lischinski, D., Farbman, Z., Uyttendaele, M. & Szeliski, R. (2006). Interactive local adjustment of tonal values. *Transactions on Graphics*, **25**, 646 – 653. 135, 172

Mann, S. & Picard, R.W. (1995). Extending dynamic range by combining differently exposed pictures. In *Society for Imaging Science and Technology* 48^{th} *anual conference*. 129

Martin, D., Fowlkes, C., Tal, D. & Malik, J. (2001). Database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *8th International Conference on Computer Vision*. x, 102, 104, 105

Meila, M. & J, S. (2001). A random walks view of spectral segmentation. In *AI and Statistics*. 26, 45, 70, 81, 86, 97

Mitsunaga, T. & Nayar, S.K. (2000). High dynamic range imaging: Spatially varying pixel exposures. In *Computer Vision and Pattern Recognition*. 128, 129

Mortensen, E.N. & Barrett, W.A. (1995). Intelligent scissors for image composition. In *SIGGRAPH*. 43, 147

Morton, K.W. & Mayers, D.F. (2005). *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press. 52, 59, 136

Mumford, D. & Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications On Pure & Applied Mathematics*, **42**, 577 – 685. 14, 39, 43

Nadler, B., Lafon, S., Coifman, R.R. & Kevrekidis, I.G. (2005). Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems*. 15, 69, 70, 81

Nakahara, M. (1990). *Geometry, Topology and Physics*. Adam Hilger. 22

Newman, M.E.J. (2005). A measure of betweenness centrality based on random walks. *Social Networks*, **27**, 39 – 54. 90

Ng, A.Y., Zheng, A.X. & Jordan, M.I. (01). Link analysis, eigenvectors and stability. In *17th international joint conference on Artificial intelligence*. 82

Ng, M.K. (1995). Nonlinear image restoration using fft-based conjugate gradient methods. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol.2)-Volume 2*, 2041, IEEE Computer Society, Washington, DC, USA. 53, 54

Nocedal, J. & Wright, S.J. (1999). *Numerical Optimization*. Springer Verlag. 171

Oh, B.M., Dorsey, J. & Durand, F. (2001). Image-based modeling and photo editing. *ACM SIGGRAPH*, 433 – 442. 128

Osher, S. & Fedkiw, R. (2003). *Level Set Methods and Dynamic Implicit Surfaces*. Springer. 14, 40, 136

Osher, S. & Sethian, J. (1988). Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, **79**, 12–49. 14, 40

Page, L., Brin, S., Motwani, R. & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford University. 8, 23, 65, 69, 70, 90

Papandreou, G. & Maragos, P. (2007). Multigrid geometric active contour models. *IEEE Transactions on Image Processing*, **16**, 229–240. 42

Paris, S. & Durand, F. (2008). A topological approach to hierarchical segmentation using mean shift. In *IEEE International conference on Computer VIsion and Pattern Recogntion*. 83, 134

Paris, S. & Durand, F. (2009). A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, **81**, 24 – 52. 115, 170

Pattanaik, S.N., Ferwerda, J.A., Fairchilf, M.D. & Greenberg, D.P. (1998). A multiscale model of adaptation and spatial vision for realistic image display. In *ACM SIGGRAPH*. 132

Perona, P. & Malik, J. (1990a). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629 – 639. 3

Perona, P. & Malik, J. (1990b). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**, 629 – 639. 10, 18, 19, 35, 66, 125, 130

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., H, H. & Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, **23**, 664 – 672. 115, 124, 128, 129, 130, 142

Peyré, G., Bougleux, S. & Cohen, L. (2008). Non-local regularization of inverse problems. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*. 15

Poggio, T., Torre, V. & Koch, C. (1985). Computational vision and regularization theory. *Nature*, **317**, 314 – 319. 2

Polyakov, A.M. (1981). Quantum geometry of bosonic strings. *Physics Letters B*, **103**, 207 – 210. 22

Portilla, J. & Simoncell, E.P. (2000). Image denoising via adjustment of wavelet coefficients magnitude. In *7th International Conference on Image Processing*. 36

Portilla, J., Strela, V., Wainwright, M.J. & Simoncell, E.P. (2003). Image denoising using scale mixtures of gaussians in the wavelet deomain. *IEEE Transactions on Image Processing*, **12**, 1338 – 1351. 36, 102, 115

Pothen, A., Simon, H. & Liou, K.P. (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis Applications*, **11**, 430 – 452. 15

Roth, S. & Black, M.J. (2009). Fields of experts. *International Journal of Computer Vision*, **82**, 205 – 229. 15, 29, 30, 35, 76, 78, 125

Rother, C., Komolgorv, V. & Blake, A. (2004). Grabcut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, **23**, 309 – 314. xv, 43, 46, 64, 148, 153, 154, 155, 157, 161

Rowels, S.T. & Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**, 2323 – 2326. 170

Rudin, L.I., Osher, S. & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, **60**, 259–268. 14, 35

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM. 59, 108

Schmidt, W., Gao, Q. & Roth, S. (2010). A generative perspective on mrfs in low-level vision. In *IEEE International Conference on Computer Vision and Pattern Recognition*. 76, 77, 115

Sethian, J. (1996). *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press. 136

Sharon, E., Galun, M., Sharon, D., Basri, R. & Brandt, A. (2006). Hierarchy and adaptivity in segmenting visual scenes. *Nature*, **42**, 810 – 813. 45

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 888 – 905. 5, 6, 15, 23, 35, 39, 43, 44, 67, 80, 81, 86, 97, 170

Simoncell, E.P. (1999). Bayesian denoising of visual images in the wavelet domain. *Lecture Notes in Statistics*, **141**, 613 – 627. 36

Singaraju, D. & Vidal, R. (2008). Interactive image matting for multiple layers. In *IEEE Conference on Computer Vision and Pattern Recognition*. 148

Sochen, N., Kimmel, R. & Malladi, R. (1995). From high energy physics to low level vision. Tech. rep., Lawrence Berkeley National Laboratory UCLA. 4, 21, 22, 35

Stark, J.L., Cande E, J. & Donoho, D.L. (2002). The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, **11**, 679 – 684. 37

Steidl, G., Weickert, J., Brox, T., Mrzek, P. & Welk, M. (2004). On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM Journal on Numerical Analys*, **42**, 686 – 713. 37

Strang, G. (1986a). *Introduction to Applied Mathematics*. 53, 54

Strang, G. (1986b). A proposal for toeplitz matrix calculations. *Studies in Applied Mathematics*, **2**, 171–176. 53, 54

Strang, G. (1999). The discrete cosine transform. *SIAM Review*, **41**, 135–147. 53, 54

Strümer, M., Köstler, H.&Rüde, U. (2008). A fast full multigrid solver for applications in image processing. *Numerical Linear Algebra*, **15**, 187–200. 42, 66

Sumengen, B. & Manjunath, B.S. (2006). Graph partitioning active contours (gpac) for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 509 – 521. 15

Szeliski, R. & Terzopoulos, D. (1989). Parallel multigrid algorithms and computer vision applications. In *Proceedings of the 4th Copper Mountain Conference on Multigrid Methods*. 42

Szummer, M. & Jaakkola, T. (2002). Partially labeled classification with markov random walks. *Advances in Neural Information Processing System*, **14**, 945 – 952. 70

Tenenbaum, J., Silva, A. & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reductio. *Science*, **5500**, 2319 – 2323. 170

Tomasi, C. & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision*. 26, 36, 67, 115, 130, 134

Trefethen, L.N. & Embree, M. (2005). *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press. 86

Tsuzurugi J, O.M. (2002). Statistical mechanics of the bayesian image restoration under spatially correlated noise. *Physica Review E*, **66**. 35

Tumblin, J. & Hodgins, B.K., J. K Guenter (1999). Two methods for display of high contrast images. *ACM Transactions on Graphics*, **18**, 56 – 94. 129, 132

Vogel, C. & Oman, M.E. (1996). Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, **17**, 227–238. 66

Wang, Z. (????). The ssim index for image quality assessment. [online]. http://www.ece.uwaterloo.ca/ z70wang/research/ssim/. 107

Wang, Z., Bovik, A.C., Sheikh, H.R. & Simoncell, E.P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, **13**, 600 – 612. 107

Weickert, J. (1998). *Anisotropic Diffusion in Image Processing*. ECMI Series, Teubner-Verlag, Stuttgart, Germany. 20

Weickert, J., Ter Haar Romeny, B.M. & A, V.M. (1998). Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE transactions on image processing*, **7**, 398–410. 66, 125

Weickert, J., Ishikawa, S. & Imiya, A. (1999). Linear scale-space has first been proposed in japan. *Journal of Mathematical Imaging and Vision*, **10**, 237 – 252. 18, 35

Welling, M., Hinton, G.E. & Osindero, S. (2002). Learning sparse topographic representations with products of student-t distributions. In *Advances in Neural Information Processing Systems*. 30, 76

Winnemller, H., Olsen, C. & Gooch, B. (2006). Real-time video abstraction. In *ACM SIGRAPH*. 118, 120, 132, 134, 145

Witkin, A. (1983). Scale-space fltering. In *International Joint Conference Artificial Intelligence*. 18

Wu, Z. & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1101 – 1113. 44

Xu, C. & Prince, J.L. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, **7**, 359–369. 14, 41, 48, 49, 52, 53

Yan, S. & Wang, H. (2009). Semi-supervised learning by sparse representation. In *SIAM International Conference on Data Mining*. 171

Yan, S., Zhang, B., Zhang, H.J., Yang, Q. & Lin, S. (2007). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 40 – 51. 171

Yang, A.Y., Ganesh, A., Sastry, S.S. & Ma, Y. (2010). Fast l_1 minimization algorithms and an application in robust face recognition: A review. Tech. rep., EECS Department, University of California, Berkeley. 32

Yedidia, J.S., Freeman, W.T. & Weiss, Y. (2003). *Exploring artificial intelligence in the new millennium*, chap. 8, Understanding belief propagation and its generalizations, 239 – 236. Morgan Kaufmann. 16, 78

Yu, Z. & Bajaj, C. (2002). Normalized gradient vector diffusion and image segmentation. *Computer Vision ECCV 2002, Lecture Notes in Computer Science*, **2352/2002**, 63 – 82. 41

Zhang, F. & Hancock, E.R. (2006). Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, **41**, 3328 – 3342. 15, 26, 45, 67, 70, 81, 86, 126

Zhou, D. & Schlkopf, B. (2004). A regularization framework for learning from graph data. In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*. 15, 67, 68, 69

Zhou, D. & Schlkopf, B. (2005). Regularization on discrete spaces. In *Pattern Recognition, 27th DAGM Symposium*. 67

Zhou, D., Bousquet, O., Navin, T., Weston, L., Weston, J. & Schlkopf, B. (2003). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, **16**, 321 – 328. 67, 68, 71, 90, 91

Zhou, D., Bousquet, O., Lal, T.N., Weston, J. & Schlkopf, B. (2004a). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, **16**, 321 – 328. 69

Zhou, D., Weston, J., Gretton, A., Bousquet, O. & Schlkopf, B. (2004b). Ranking on data manifolds. *Advances in Neural Information Processing Systems*, **16**, 169 – 176. 69, 90, 91

Zhou, D., Schölkopf, B. & Hofmann, T. (2005). Semi-supervised learning on directed graphs. *Neural Information Processing Systems*, **17**, 1633 –1640. 79

Zhu, S.C., Wu, Y.N. & Mumford, D. (1998). Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, **27**, 847 – 856. 125

Zhu, X. & Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Tech. rep., Technical Report CMU-CALD-02-107, Carnegie Mellon University. 90, 91

Zhu, X., Ghahramani, Z. & J., L. (2003a). Semi-supervised learning using gaussian fields and harmonic function. In *International Conference on Machine Learning*. 10, 69, 86, 126, 150, 152, 157

Zhu, X., Lafferty, J. & Ghahramani, Z. (2003b). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. 170