Image Denoising with Variational Methods via Graph Cuts

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Lukas Lang
Matrikelnummer 0504043

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:	Univ.-Prof. Dr. Günther Raidl
Mitwirkung:	Univ.-Prof. Dr. Otmar Scherzer
	Univ.-Prof. Dr. Monika Henzinger

Wien, 21. Dezember 2011 ____________________ ____________________
(Unterschrift Verfasser) (Unterschrift Betreuung)

Image Denoising with Variational Methods via Graph Cuts

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computational Intelligence

by

Lukas Lang
Registration Number 0504043

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.-Prof. Dr. Günther Raidl
Assistance: Univ.-Prof. Dr. Otmar Scherzer
Univ.-Prof. Dr. Monika Henzinger

Vienna, 21. Dezember 2011 ____________________ ____________________
(Signature of Author) (Signature of Advisor)

Acknowledgements

First, I want to thank my supervisors Monika Henzinger and Otmar Scherzer for inviting me to work with them, for sharing their expertise and for their valuable advices. Moreover, they have provided me with resources and placed me in a very valuable environment.

Likewise, I want to thank Clemens and Thomas for their helpful discussions and comments not only in mathematical matters. Moreover, I thank Günther Raidl for supporting this joint work and his group for triggering my interest in algorithms and complexity.

I want to thank my colleagues from university and especially my friends for the entertaining moments, their caring and the enjoyable time we spent. For the daily support I feel grateful to Marlene. Finally, I am deeply grateful to my parents for their lifelong support and for encouraging me in my academic studies.

Abstract

Graph cut methods have evolved to a well-investigated and acknowledged method in computer vision. They have successfully been applied to a great variety of applications such as medical image processing, image restoration and segmentation, and many more. Many problems in computer vision arise from the need of determining the *maximum a posteriori* estimate in a stochastic *Markov random field* model, which in fact is equivalent to minimizing some energy function. These energies incorporate on the one hand the deviation from observed data and on the other hand the smoothness characteristics of the solution.

For a certain type of energy functions, graph cuts provide a novel way to exactly infer the maximum a posteriori estimate by computing a minimum cut. The energies are modeled as flow networks and due to the important *max-flow-min-cut theorem* the minimum cut can be found efficiently by computing the maximum flow. However, as soon as such energy function increases in complexity, either by extending the range of labels (multilabel problem) or by adding complex interaction potentials, the problem of inferring the exact MAP estimate becomes NP-hard.

Especially the subject of image denoising, which is the reconstruction of an image that has been degraded by noise, has received extensive attention from the image analysis community. Several continuous regularization methods for denoising have been proposed. In the course of this work we investigate the applicability of graph cut methods and approximations for image denoising. In particular, we study discrete forms of *first-order regularization* models. Moreover, on the basis of test images which were artificially degraded by (e.g. Gaussian) noise we conduct a series of experiments with known graph constructions and show that even complex energy functions can be approximated with sufficient quality.

Kurzfassung

Graph Cut basierte Methoden wurden in den letzten Jahren zu einem gut erforschten und anerkannten Verfahren im Bereich Computer Vision entwickelt. Eine Vielzahl von Anwendungen in den Bereichen medizinische Bildverarbeitung, Bildrestaurierung und Bildsegmentierung wurden erfolgreich auf Basis des Verfahrens umgesetzt.

Üblicherweise werden Probleme im Umfeld der Bildverarbeitung als stochastische Markov Random Field Modelle formuliert um aussagekräftige Schlüsse über verborgene Information ziehen zu können. Die Inferenz in solchen Modellen ist äquivalent zur Minimierung einer Energiefunktion, welche einerseits die Abweichung einer Lösung zu den beobachteten Daten und andererseits die Gleichmäßigkeit der Lösung wiederspiegelt.

Gewisse Klassen von Energiefunktionen können mit Graph Cut Methoden exakt minimiert werden sofern die Energiefunktion in einem Graphen repräsentiert werden kann. Auf Grund der berühmten Max-Flow-Min-Cut Äquivalenz kann ein minimaler Schnitt effizient mittels maximalen Fluss in einem Fluss Netzwerken berechnet werden. Leider können nur sehr einfache Energiefunktionen exakt in polynomieller Zeit minimiert werden.

Besondere Aufmerksamkeit hat die Aufgabe des Entrauschens von Bildern eingenommen. Hierbei wird versucht ein durch Rauschen gestörtes Bild so gut als möglich wiederherzustellen. Im Zuge dieser Arbeit untersuchen wir die Anwendbarkeit von Graph Cut Methoden hinsichtlich verschiedener konvexer Regularisierungsmodelle erster Ordnung und führen für eine ausgewählte Anzahl an diskreten Energien Experimente auf Basis von künstlich verrauschten Bildern durch. Aufgrund der Komplexität der vorgestellten Modelle können diese nur approximiert werden. Jedoch sind die gewonnenen Ergebnisse von ausreichender Qualität.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Over the last years, graph cut methods have evolved to a well-investigated and established method in computer vision. Graph cuts have successfully been applied to a great variety of applications such as medical image processing, image restoration and segmentation, and many more.[1] Frequently, these problems are posed as abstract classification or (pixel) labeling problems in which one has to assign each object a label with the objective of minimizing the total assignment cost.

Many problems in computer vision arise from the need of determining the *maximum a posteriori* (MAP) estimate in a stochastic *Markov random field* (MRF) model, which in fact, is equivalent to minimizing some energy function [84] that incorporates on the one hand the deviation from an observed data, and on the other hand the smoothness characteristics of the solution.

For a certain type of discrete energy functions, graph cuts provide a novel way to exactly infer the MAP estimate by computing a minimum cut [25, 30, 56, 76]. The energies are modeled as the capacities of a flow network and due to the important *max-flow-min-cut theorem* [40, 41] the minimum cut can be found efficiently by computing the maximum flow. Several (strongly) polynomial-time algorithms have been proposed for computing the maximum flow and equivalently the minimum cut.[2]

However, as soon as such an energy function increases in complexity, either by extending the range of labels (multilabel problem) or by adding complex interaction potentials, the problem of inferring the exact MAP estimate becomes NP-hard [19, 76]. Nevertheless, several approximations have been proposed [19, 22, 63, 66, 72, 83].

[1] We refer the reader to Szeliski et al. [109] for a recent survey on methods for energy minimization in computer vision and `http://vision.middlebury.edu/`, which lists the current benchmark results.

[2] See e.g. Goldberg and Tarjan [49] for a survey.

1.2 Problem Statement

However, especially the subject of image denoising, which is the reconstruction of an image that has been degraded by noise, has received extensive attention from the image analysis community. Several continuous regularization methods for denoising have been proposed. In the course of this work, we investigate the applicability of graph cuts for image denoising. Special attention will be given to continuous regularization functionals and in particular the applicability of graph cut methods to *anisotropic first-order regularization* will be examined. We establish finite-dimensional discrete forms of the continuous regularization functionals and discuss the applicability of known graph constructions. Moreover, experimental results are presented.

1.3 Previous and Related Work

Historically, the labeling problem was first considered by Stone [108] in the context of distributed computing, where computational tasks had to be scheduled for computation on multiple processors. For two labels, Stone computed the global minimum of the cost function with a graph cut. In a first application in the context of image processing, Greig et al. [50] successfully reduced the problem of image denoising of black and white images to the minimum cut problem. Almost ten years later, Boykov et al. [19] further investigated graph cuts and approximations. Moreover, Kleinberg and Tardos [65] were the first to give a context-independent formulation of the problem and developed several approximation algorithms based on linear program relaxations and randomized rounding schemes. Later, Veksler [111] and Boykov et al. [20, 21, 22] developed further graph cut-based approximations: the *expansion algorithm* and the *swap algorithm*.

In their work, Kolmogorov and Zabih [75, 76] stated a necessary and sufficient condition for the exact minimization of Boolean energy functions arising from first- and second-order Markov Random Fields. Later, Freedman and Drineas [42], Freedman and Turek [44] gave an algebraic characterization of the results of Kolmogorov and Zabih, established the connection to *pseudo-Boolean functions*, and further extended the class of energy functions which can be minimized exactly in polynomial time. Pseudo-Boolean functions appear in combinatorial optimization and have been studied for more than fifty years [11, 28, 61, 91].

For a linearly ordered label set, Ishikawa [56] and Darbon [30] showed how to exactly compute the minimum of energy functions with convex, and with submodular pairwise interaction terms, respectively. Recently, Charpiat [25] suggested a graph construction which strictly extends the class of functions which can be minimized exactly via graph cuts and for the first time includes some nonsubmodular energies.

Lempitsky et al. [83] later generalized the approximations by Boykov et al. [22] to *fusion moves*. Moreover, Kolmogorov and Rother [72] and Rother et al. [98, 99] investigated a method from the domain of (quadratic) pseudo-Boolean function minimization, the so called *roof duality*, which allows the computation of a partially optimal solution in case of nonsubmodular functions and proved useful in many applications (e.g. Woodford et al. [117]). The concept was introduced by Hammer et al. [52] and has been known for quite some time. Aside from linear programming-based methods, Boros et al. [12][3] proposed a network flow-based method, the so called *BHS algorithm*[4], for efficiently computing such a partial solution.

Applications

As graph cut methods have gained great attention over the last years, the number of publications in journals and conference proceedings raised to a tremendous level.[5] However, we give a brief overview of its main applications.

One of the earliest problems in computer vision which was approached with graph cuts is the task of *image restoration* [5, 22, 30, 50, 60, 111, 119]. The goal is to restore an image which was somehow degraded. If the corruption is just by noise, we speak of *image denoising*, which is the main area of applications of graph cuts. Other restoration problems aside from denoising are *image deblurring* and *inpainting*, which are not considered here.

The task of *image segmentation* [9, 14, 15, 17, 27, 37–39, 51, 58, 77, 95, 97, 107, 112, 114, 118] is to partition a given image into segments such as foreground and background, or more generally into multiple (disjunct) segments such as objects visible in an image.

Stereo and *multiview reconstruction* [8, 19, 22, 59, 64, 71, 73, 74, 74, 100, 102, 111] aim at determining the correspondence between pixels in multiple images. For instance, in the *two-camera stereo problem* one needs to find the correspondence between pixels in a static scene taken from two horizontally shifted cameras. Then, a 3D model can be constructed from the correspondence.

Another major task is *motion estimation* [22, 26, 43, 82, 83, 115] which for instance is used to estimate an *optical flow*, i.e. the pixel movements, in a sequence of images. The difference to stereo or multiview reconstruction is that in general both the camera and the objects may move within a sequence of images.

Moreover, we shall mention successful graph cut applications in *texture synthesis* [79] and *digital photomontage* [1, 96]. In texture synthesis one needs to pursue the structures of a texture for instance to fill up missing or deleted parts of an image. Digital photomontage includes for instance the task of stitching multiple images such that the boundaries are as smooth as possible.

1.4 Contribution

In the course of this work, a literature study on the topic of graph cut methods, the theoretic foundations, and its applications is done. Then, continuous first-order regularization functionals for denoising are brought into a finite-dimensional discrete form and, moreover, we investigate the applicability of the discussed graph cut methods. Special attention will be given to *anisotropic*

[3]Unfortunately, we were not able to obtain the original publication. However, the main concepts are covered by Boros and Hammer [11].

[4]Often this method is incorrectly referred to as *quadratic pseudo-Boolean optimization* (QPBO), which rather denotes the methods used for the minimization/maximization of pseudo-Boolean functions of degree at most two. According to Blake et al. [10] the term *BHS algorithm* is more appropriate.

[5]A great variety of publications can be found at `http://www.cvpapers.com/` and `http://muq.org/~cynbe/vtopics.html#Graph_Cut`.

first-order regularization. Finally, we present and discuss experimental results obtained by an implementation based on existing libraries.

1.5 Structure of the Thesis

The remainder of this thesis is structured as follows. In Chapter 2, we introduce basic image and noise models. Moreover, we discuss the relation between maximum flows and minimum cuts in flow networks. Chapter 3 is devoted to Markov random fields and the Bayesian justification of energy minimization. In Chapter 4, we present the standard graph construction for the minimization of MRF energies, approximation algorithms, and extensions. Moreover, we discuss continuous total variational methods for image denoising, investigate the applicability of graph cuts, and in Chapter 5 we present our experimental results. Chapter 6 concludes this thesis.

1.6 Notational Conventions

$\mathbb{B}$	The set $\{0, 1\}$.
$\mathbb{N}$	The set of natural numbers.
$\mathbb{R}$	The set of reals.
J_n	An $n \times n$ matrix of ones.
VERTEX COVER	A problem or formal language.
P, NP	Complexity classes.
$\mathcal{G}$	A graph.
$\mathcal{V}$	The set of vertices/nodes.
$\mathcal{E}$	The set of edges/arcs.
$c(u, v)$	The capacity of an edge $\{u, v\} \in \mathcal{E}$ or arc $(u, v) \in \mathcal{E}$.
$c(S, T)$	The capacity of a cut (S, T), where $S, T \subset \mathcal{V}$.
Ω	The image area.
n_x	The number of horizontal nodal points.
n_y	The number of vertical nodal points.
u	An intensity function of a continuous image defined on Ω.
∇u	The gradient of u.
$\mathbf{u}$	A matrix of a discrete image.
u_x, u_y	The discrete gradients of a discrete image $\mathbf{u}$ along x and y.
$\mathbf{X}$	A vector of hidden variables in a Markov random field.
$\mathbf{Z}$	A vector of observation variables in a Markov random field.
$\mathbf{x}$	A vector of realizations of $\mathbf{X}$.
$\mathbf{z}$	A vector of realizations of $\mathbf{Z}$.
$\mathbf{x}^c$	The transition function in move-making algorithms.
$\mathcal{L}$	The space of labels (label set).
Φ_i	Unary energy potential of variable x_i.
Ψ_{ij}	Pairwise energy potential of related variables x_i, x_j.
Ψ_C	Energy potential of a clique C of variables.
α	The regularization parameter.

CHAPTER 2

Preliminaries

2.1 Image and Noise Models

In this section, we introduce discrete (digitized) and continuous images. Moreover, we describe noise models, which account for various types of errors in the recorded images. Most important for our considerations are errors in intensity, which often result from noise interfering with image sensors in digital cameras or scanners.

Discrete Images

In this subsection, we will establish a basic model of discrete and continuous images as in Scherzer et al. [103].

Let $h > 0$ and let $n_x, n_y \in \mathbb{N}$. A two-dimensional *discrete image* of the size $n_x \times n_y$ is given as a matrix $\mathbf{u} = (u_{ij})_{(i,j)\in\mathcal{I}_1}$, where

$$u_{ij} \in \mathbb{R}, \quad (i,j) \in \mathcal{I}_1 := \{1,\dots,n_x\} \times \{1,\dots,n_y\}.$$

The values u_{ij} are the intensity values at the nodal points $x_{ij} = (ih, jh)$, where $(i,j) \in \mathcal{I}_1$. These nodal points are aligned along a rectangular *pixel grid* $\mathbf{x} = (x_{ij})$, which is assumed to be regular. The parameter h controls the resolution of a discrete image, which is the horizontal and vertical distance between the pixels x_{ij}. Figure 2.1 illustrates the described setting (taken from Scherzer et al. [103]).

In addition, let us assign to every pair

$$(i,j) \in \mathcal{I}_2 := \{1,\dots,n_x - 1\} \times \{1,\dots,n_y - 1\}$$

the *discrete gradient* v_{ij} of $\mathbf{u}$ at node x_{ij} defined as

$$v_{ij} := \frac{1}{h}\begin{pmatrix} u_{i+1,j} - u_{ij} \\ u_{i,j+1} - u_{ij} \end{pmatrix} := \begin{pmatrix} u_x \\ u_y \end{pmatrix}.$$

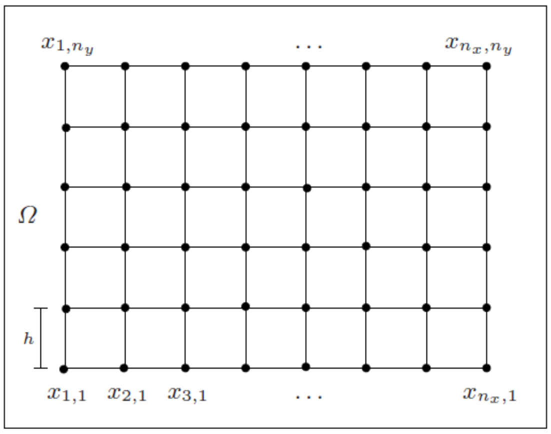


Figure 2.1: A regular pixel grid with nodal points $x_{ij} = (ih, jh)$.

The gradient is a measure for the change in intensity along the directions of x and y.

In contrast, a *continuous image* is given by its intensity function $u : \Omega \to \mathbb{R}$, where

$$\Omega := (0, (n_x + 1)h) \times (0, (n_y + 1)h)$$

is the image area. It is noteworthy that Ω is chosen such that the entire pixel grid $\mathbf{x}$ is contained in Ω. The gradient of a continuous image u is denoted by

$$\nabla u := \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix}.$$

In the course of this work we will deal only with discrete images and thus approximate the continuous gradient ∇u by the discrete gradient.

Noise Models

In this subsection, we discuss various noise models which characterize the distortions arising from image recording. Often, noise origins from image sensors or analog-to-digital converters in cameras or scanners and results in different type of distortions.

Most important for our purpose are errors in intensity, which manifest in a local character. It is assumed that the observed errors are realizations of independent and identically distributed (i.i.d.) random variables. Another form of distortion are sampling errors, where the observed error also depends on the intensity of the surrounding area.

Intensity Errors

The most basic noise model regarding errors in intensity is *additive noise*. Given a discrete image $\mathbf{u}$ and an $n_x \times n_y$ matrix $\boldsymbol{\delta} = (\delta_{ij})_{(i,j)\in\mathcal{I}_1}$ of realizations of i.i.d. random variables, we speak of *additive intensity errors* if the recorded or observed data are

$$\mathbf{u}^{\delta} = \mathbf{u} + \boldsymbol{\delta}.$$

In case that the random variables obey a Gaussian distribution, $\mathbf{u}^{\delta}$ is said to contain *Gaussian intensity errors*. For instance, thermal noise is Gaussian white noise, i.e. independently distributed additive noise with zero mean and variance σ^2 [116]. Other common distributions used in additive noise models are Laplacian and Poisson distributions.

Moreover, if the noise cannot be described by the above additive model but as a function $\boldsymbol{\delta}$ of the original image $\mathbf{u}$, the observed data is stated as

$$\mathbf{u}^{\delta} = \boldsymbol{\delta}(\boldsymbol{u}).$$

Prominent noise models which employ such functional dependencies are *Poisson noise* and *Salt & Pepper noise*. The former is used to model photon counting errors produced by charge-coupled device (CCD) sensors, where $\delta_{ij}(u_{ij})$ denotes the number of photons detected and is treated as the realization of a Poisson distributed random variable with mean u_{ij}. The latter assumes that there exists a lower bound c_{min} and an upper bound c_{max} on the values of $\mathbf{u}$ such that $c_{min} \leq u_{ij} \leq c_{max}$. Then, Salt & Pepper noise sets the intensity of each pixel u_{ij} to one of $\{c_{min}, u_{ij}, c_{max}\}$ according to some probability distribution defined on this set.

Figure 2.2 illustrats (artificial) additive Gaussian noise (2.2(b)), Poisson noise (2.2(c)), and Salt & Pepper noise (2.2(d)) applied to a greyscale image (2.2(a)).

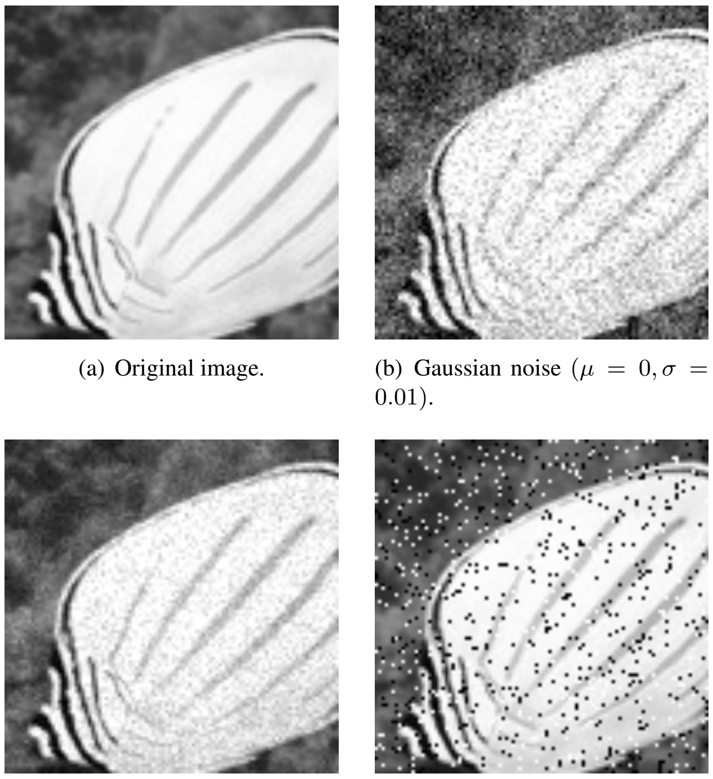
Example 1. *Consider for instance the discrete 8-bit greyscale image in Figure 2.2(d). Then,* $c_{min} = 0$ *corresponds to the color black and* $c_{max} = 255$ *corresponds to white. The depicted image has been degraded by Salt & Pepper noise such that for each pixel the original intensity* u_{ij} *is replaced by the colors black or white with probability* 0.1.

2.2 Network Flows and Minimum s-t Cuts

In this section, we give a brief introduction to networks flows, cuts, and algorithms for the efficient computation.

A very broad class of problems can be modeled as a *transportation* or *flow network*. Informally, such a network carries some kind of "traffic" via its nodes and arcs from a defined source "producing" the traffic to a defined sink "consuming" it. In addition, the transportation network consists of intermediate nodes acting as "switches" passing the traffic to adjacent nodes. Each arc is associated with a certain capacity limiting the maximum amount of traffic being forwarded to its incident node. For a better understanding, imagine for example a pipeline or highway network. For simplicity it is assumed that the source has unlimited supply and the sink is able to absorb an unlimited amount of traffic. In the following, we give a precise formulation of the described setting.

Definition 1. *A* ***graph*** *is an ordered pair* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $\mathcal{V}$ *denotes a set of nodes and* $\mathcal{E}$ *is a collection of edges.*


(a) Original image. (b) Gaussian noise ($\mu = 0, \sigma = 0.01$).

(c) Poisson noise. (d) Salt & Pepper noise ($p = 0.1$).

Figure 2.2: Corrupted grayscale images.

We speak of an *undirected* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if the elements in $\mathcal{E}$ are two-element subsets of $\mathcal{V}$: $\{u, v\}$ for some $u, v \in \mathcal{V}$. Conversely, a *directed* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}')$ consists of nodes $\mathcal{V}$ and arcs (or directed edges) $\mathcal{E}' \subseteq \mathcal{V} \times \mathcal{V}$. The elements in $\mathcal{E}'$ are ordered pairs (u, v).

Definition 2. *A **flow network** $(\mathcal{G}, c)$ is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where a nonnegative **capacity** $c(u, v) \geq 0$ is associated with each arc $(u, v) \in \mathcal{E}$.*

Furthermore, $\mathcal{V}$ contains two special nodes: a *source* s and a *sink* t, also referred to as *terminals*. Node s is the source of all "traffic" whereas node t consumes it. Therefore, we require that no arc enters the source and no arc leaves the sink. To yield a valid flow network, we demand that every node $v \in \mathcal{V}$ is on some path from s to t and no path of infinite capacity exists from s to t. Finally, we disallow loops, i.e. $(u, u) \notin \mathcal{E}$. Figure 2.3 depicts a simple flow network with four nodes and five arcs and capacities written next to the arcs.

Definition 3. *An **s-t flow**[1] in a network $(\mathcal{G}, c)$ is a function $f : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}^+$ that satisfies the following two constraints:*

[1]We will use the terms *s-t flow* and *flow* synonymously since s-t flows are the only flows we are interested in.

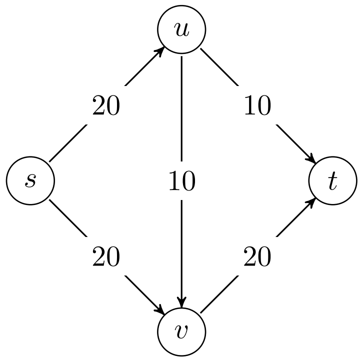


Figure 2.3: A flow network.

a) (capacity constraint) for every pair $u, v \in \mathcal{V}$,

$$0 \leq f(u,v) \leq c(u,v). \tag{2.1}$$

b) (flow conservation) for each node $u \in \mathcal{V} \setminus \{s,t\}$,

$$\sum_{v \in \mathcal{V}} f(v,u) = \sum_{v \in \mathcal{V}} f(u,v). \tag{2.2}$$

The first constraint ensures that the flow on every arc is nonnegative and within the corresponding capacity. The second constraint states that, except for the source and the sink, all flow must be forwarded. In other words, intermediate nodes can neither "store" nor "create" flow. Thus, it is quite natural to ask for the amount of flow carried by a network.

Definition 4. *The **value** $|f|$ of a flow f is defined as*

$$|f| = \sum_{v \in \mathcal{V}} f(s,v). \tag{2.3}$$

An immediate question is to determine the maximum amount of flow that can be sent trough a given flow network, which leads to the MAXIMUM-FLOW problem:

Instance: A flow network $(\mathcal{G}, c)$ with a single source s and a single sink t.
Question: What is the maximum feasible flow f in $\mathcal{G}$ from s to t?

Ford and Fulkerson [41] were the first to publish an algorithm based on augmenting paths, which runs in time $O(|\mathcal{E}| \cdot \max|f|)$. Several other (strongly) polynomial-time algorithms and variations thereof such as the *Preflow-Push Algorithm*, which takes time $O(|\mathcal{V}|^2 \cdot |\mathcal{E}|)$, exist to compute a maximum flow. Various improvements have been made to these algorithms.[2]

It is easy to see that the sum of the capacities of the arcs leaving the source and the sum of the capacities of the arcs entering the sink are trivial upper bounds for the maximum flow that

[2]We refer to Ahuja et al. [2] for the general topic of network flows.

can be sent along a network. However, the *max-flow min-cut* theorem states that the maximum flow is equal to the capacity of a minimum cut. In order to formulate the important theorem we first need to introduce the notion of an *s-t cut*:

Definition 5. *An* ***s-t cut***[3] (S, T) *of a flow network* $(\mathcal{G}, c)$ *is a partition of the vertices* $\mathcal{V}$ *into* S *and* $T = \mathcal{V} \setminus S$ *such that* $s \in S$ *and* $t \in T$*. The* ***capacity*** *or* ***cost*** $c(S, T)$ *of a cut* (S, T) *is defined as*

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v). \tag{2.4}$$

A *minimum cut* of a network $\mathcal{G}$ is a cut whose capacity is the minimum among all cuts of the network. It is easy to see that by deleting all edges directed from S to T the terminals get disconnected. For convenience, we call this set $\{(u, v) \in \mathcal{E} \mid u \in S, v \in T\}$ the *cut-set*. We are now ready to state the important theorem proved independently by Elias et al. [40], and by Ford and Fulkerson [41]:

Theorem 1 (Max-flow Min-cut)**.** *In every flow network, the maximum value of a flow is equal to the minimum capacity of a cut.*

The importance of the equivalence lies in the ability to efficiently find a minimum cut of a network by computing the maximum flow. We shall point out that other (efficient) algorithms exist for finding minimum cuts.

[3]In the following we will use the terms *s-t cut*, *graph cut*, and *cut* synonymously since we only consider single source and a single sink flow networks.

CHAPTER 3

Markov Random Fields

In this chapter, we discuss one of the fundamental stochastic models used in computer vision, namely Markov Random Fields (MRF). They provide a convenient way for modeling image properties with contextual constraints with the primary goal of making inferences about images. Its applications range from image reconstruction and denoising to image segmentation, 3D vision, and object labeling. Moreover, MRFs provide insight into the computational aspect of deriving the joint probability of an image and of inferring the Maximum a Posterior (MAP) estimate given some observation. In this chapter we develop a Bayesian justification for energy minimization in order to derive the MAP estimate. Under certain conditions, these energies can then be minimized or approximated efficiently with graph cut methods as we will see.

3.1 Hidden Markov Models

An important concept are the so called *Hidden Markov Models* (HMM), which basically consist of n hidden random variables denoted by a vector $\mathbf{X} = (X_i)_{1 \leq i \leq n}$. These variables account for observable quantities (e.g. in image analysis the intensity measured at some pixel) but cannot directly be observed. The fundamental idea is to assume that the observations or measurements $\mathbf{z} = (z_i)_{1 \leq i \leq n}$ are realizations of random variables $\mathbf{Z} = (Z_i)_{1 \leq i \leq n}$ themselves. In this way, the model incorporates measurement errors produced by sensors, etc. However, we make the assumption that our random variables are discrete and take values from the finite discrete set $\mathcal{L}$. We denote by

$$P(\mathbf{Z} = \mathbf{z} \mid \mathbf{X} = \mathbf{x}) \tag{3.1}$$

the conditional probability, often referred to as the *likelihood*, of observing $\mathbf{z}$ provided that the hidden variables are in state $\mathbf{x}$. The observation variables are now conditioned by the hidden variables. In the following, we will use for better readability $\mathbf{x}$ to denote the event $\mathbf{X} = \mathbf{x}$ and $P(\mathbf{x})$ to denote the probability $P(\mathbf{X} = \mathbf{x})$ of the event. Informally, we will also refer to $\mathbf{x}$ as a *configuration* or *labeling* of the hidden variables $\mathbf{X}$.

For discrete HMMs with a finite set of possible observations, Rabiner [93] formulates three basic problems of interest for real-world applications. Probably the most interesting one for us is an inference problem and asks for the most likely instantiation of the hidden variables $\mathbf{X}$ after having observed $\mathbf{z}$. In terms of probability, we want to find a realization $\mathbf{x}$ that maximizes the *posterior* probability $P(\mathbf{x} \mid \mathbf{z})$. Bayes' formula allows to compute the posterior probability as

$$P(\mathbf{x} \mid \mathbf{z}) = \frac{P(\mathbf{z} \mid \mathbf{x})P(\mathbf{x})}{P(\mathbf{z})}, \tag{3.2}$$

where $P(\mathbf{x})$ is called the *prior* probability. The prior captures the probability of the event of the hidden variables being in state $\mathbf{x}$ without any further knowledge and solely depends on the mutual dependence between the hidden variables. After having observed $\mathbf{z}$ we can ignore the denominator $P(\mathbf{z})$ for optimization and the problem reduces to finding the realization $\hat{\mathbf{x}}$ which maximizes

$$P(\mathbf{x} \mid \mathbf{z}) \propto P(\mathbf{z} \mid \mathbf{x})P(\mathbf{x}), \tag{3.3}$$

which is referred to as the *maximum a posteriori* (MAP) estimate.

For tractability reasons, the likelihood function is commonly assumed to arise from a product distribution of the form [6]

$$P(\mathbf{z} \mid \mathbf{x}) = \prod_i P(z_i \mid x_i). \tag{3.4}$$

This simplification implies the independence between the variables $\mathbf{Z}$ and allows the direct application of noise models as discussed in Section 2.1.[1] Note that the conditional dependency between the hidden variables still remains. Figure 3.1 illustrates the described.

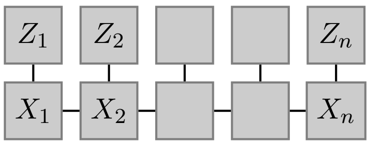


Figure 3.1: A (first-order) hidden Markov model.

3.2 Markov Random Fields

In the previous section, we have seen the relation between hidden variables and observations in (very simple) hidden Markov models. In this section, we will go into further detail and discuss the basics of Markov random fields, a crucial theoretical result—the Hammersley-Clifford theorem—which allows the efficient computation of the prior $P(\mathbf{x})$, and show how to exploit the conditional dependencies between the hidden variables to determine the Maximum a Posterior estimate.

The underlying idea of Markov random fields is to represent the conditional dependencies between hidden variables and groups thereof in an image graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of nodes

[1] We refer the reader to Scherzer et al. [103, Sec. 2.6] and Li [84, Sec. 1.3] for examples.

$\mathcal{V} = \{1, \ldots, n\}$ corresponds to the (unordered) set of pixels, which in the simplest case arises from a consecutive numbering of $n = n_x \times n_y$ nodal points in a regular grid. For a better understanding of the described consider Figure 3.2. The set $\mathcal{E}$ is the set of (undirected) edges. An edge $\{u, v\} \in \mathcal{E}$, where $u, v \in \mathcal{V}$, indicates a mutual relation between the pixels u and v. For instance, edges may indicate mutual neighborship between pixels next to each other in a regular grid.

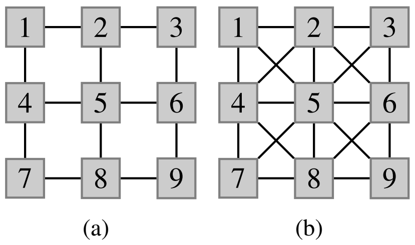


Figure 3.2: Graphs for Markov models representing image data. Figures 3.2(a) and 3.2(b) depict a regular 4-connected and a regular 8-connected pixel graph, respectively.

Above, we have already given an intuitive definition of the notion of neighborhood in a graph $\mathcal{G}$. We define it more precisely as follows:

Definition 6 (Li [84])**.** *A **neighborhood system** on the set $\mathcal{V}$ of nodes is defined as*

$$\mathcal{N} := \{\mathcal{N}_v \mid v \in \mathcal{V}\},$$

where each $\mathcal{N}_v \subset \mathcal{V}$ denotes the neighboring nodes of v, i.e. $u \in \mathcal{N}_v \Leftrightarrow \{u, v\} \in \mathcal{E}$, and fulfills the following two properties:

1. *$v \notin \mathcal{N}_v$ (i.e. a node is not a neighbor of itself), and*

2. *$v \in \mathcal{N}_u \Leftrightarrow u \in \mathcal{N}_v$ (i.e. the relationship it is mutual).*

Moreover, if C is a subset of $\mathcal{V}$, we define $\mathbf{x}_C := (x_v)_{v \in C}$. It is convenient to define the following:

Definition 7 (Li [84])**.** *A clique C for $(\mathcal{V}, \mathcal{N})$ is a subset of $\mathcal{V}$. It consists of either a single node $C = \{v\}$, a pair of neighboring nodes $C = \{v, v'\}$, a triple of neighboring nodes $C = \{v, v', v''\}$, and so on.*

Consequentially, we denote by $\mathcal{C}$ the set of all cliques in $\mathcal{G}$. For instance, the image graphs depicted in Figures 3.2(a) and 3.2(b) contain cliques of size at most two and three, respectively.

Let us now introduce the central concept of this section.

Definition 8 (Li [84])**.** *A vector of random variables $\mathbf{X}$ is said to be a **Markov random field** (MRF) on $\mathcal{V}$ w.r.t. a neighborhood system $\mathcal{N}$ on $\mathcal{V}$ if and only if the following two conditions hold:*

1. $P(\mathbf{x}) > 0$ *for all possible realizations* $\mathbf{x}$*, and*

2. $P(x_v \mid \mathbf{x}_{\mathcal{V}\setminus\{v\}}) = P(x_v \mid \mathbf{x}_{\mathcal{N}_v})$ *for all* $v \in \mathcal{V}$*.*

The first condition is introduced for pure technical reasons. The second condition is referred to as the *Markovian property* and captures the local characteristics of a Markov random field. It states that the probability of the event $X_v = x_v$ given the instantiations of all other hidden variables is equal to the probability given only the realizations of its direct neighbors. We will see that this property will be useful to quantify the prior distribution $P(\mathbf{x})$. Figure 3.3 depicts a section of a Markov random field model. The conditional dependencies are drawn as undirected edges.

Finally, the *order* of a MRF is defined as the size of the largest clique minus one.

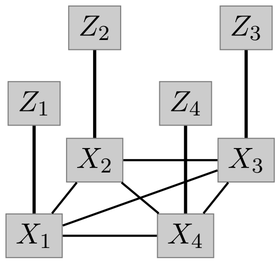


Figure 3.3: Section of a Markov random field.

Inferring the Maximum a Posterior Estimate

Recall from Section 3.1 that, given some observations $\mathbf{z}$, the Maximum a Posterior (MAP) estimate $\hat{\mathbf{x}}$ gives the most likely explanation for the measurements $\mathbf{z}$ and can be derived as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{z} \mid \mathbf{x}) P(\mathbf{x}), \tag{3.5}$$

where the likelihood function is modeled as before. The question of defining the prior distribution still remains. By the definition of the neighborhood system the conditional dependencies are cyclic (cf. Figure 3.3) and thus, the joint distribution is not straightforward to compute by factorization. Fortunately, the Hammersley-Clifford theorem provides a convenient way to specify it. Let us first define the following:

Definition 9 (Li [84])**.** *A vector of random variables* $\mathbf{X}$ *is said to be a* ***Gibbs random field*** *(GRF) on* $\mathcal{V}$ *w.r.t. a neighborhood system* $\mathcal{N}$ *if and only if the random variables obey values from a* ***Gibbs distribution****, i.e.*

$$P(\mathbf{x}) := \frac{1}{Z} e^{-E(\mathbf{x})}, \tag{3.6}$$

where Z *is the* ***partition function****, a normalizing constant defined as* $Z := \sum_{\mathbf{x}} e^{-E(\mathbf{x})}$*, and* $E(\mathbf{x})$ *is some* ***energy function****.*

The energy function is defined as the sum

$$E(\mathbf{x}) := \sum_{C \in \mathcal{C}} \Psi_C(\mathbf{x}_C) \tag{3.7}$$

of the *clique potentials* Ψ_C and runs over the set of all cliques $\mathcal{C}$. Clearly, this sum can be decomposed into sums over cliques of various sizes and the energy function can be written as

$$E(\mathbf{x}) := \sum_{C \in \mathcal{C}_1} \Psi_1(\mathbf{x}_C) + \sum_{C \in \mathcal{C}_2} \Psi_2(\mathbf{x}_C) + \sum_{C \in \mathcal{C}_3} \Psi_3(\mathbf{x}_C) + \dots, \tag{3.8}$$

where $\mathcal{C}_i$ denotes the set of cliques of size $i > 0$. We are now ready to state the following important theorem.

Theorem 2 (Besag [6], Hammersley and Clifford [53])**.** $\mathbf{X}$ *is a Markov random field on* $\mathcal{V}$ *w.r.t. a neighborhood system* $\mathcal{N}$ *if and only if* $\mathbf{X}$ *is a Gibbs random field on* $\mathcal{V}$ *w.r.t.* $\mathcal{N}$.

The crucial consequence of this theorem is that the joint probability $P(\mathbf{x})$ is explicitly given in terms of the clique potential. Moreover, the energy function incorporates prior knowledge about the model.

Equipped with this knowledge we can model the likelihood $P(z_i \mid x_i)$ just as

$$P(z_i \mid x_i) := c_i e^{-\Phi_i(x_i, z_i)}, \tag{3.9}$$

where c_i is a normalizing constant and $\Phi_i(x_i, z_i)$ describes the distance between the observation z_i and the value x_i of the hidden variable. Earlier we assumed that the observation variables are independent and thus, the likelihood $P(\mathbf{z} \mid \mathbf{x})$ can be written as a product distribution.

Based on this assumptions, it is common to directly state the *posterior MRF* in terms of the energy potential such that the energy also depends on the measurements $\mathbf{z}$. Thus, the posterior can be written as

$$P(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} e^{-E(\mathbf{x}, \mathbf{z})}, \tag{3.10}$$

where

$$E(\mathbf{x}, \mathbf{z}) = \sum_{C \in \mathcal{C}} \Psi_C(\mathbf{x}) + \sum_i \Phi_i(x_i, z_i). \tag{3.11}$$

Again, $Z(\mathbf{z})$ is the partition function which, fortunately to us, is not needed for deriving the Maximum a Posterior estimate. In fact, computing the partition function is even intractable in case the model depends on additional parameters.

The Markov-Gibbs equivalence stated by the theorem and the above assumption allow the inference of the maximum a posterior estimate $\hat{\mathbf{x}}$. Taking the logarithm of equation (3.10) yields

$$E(\mathbf{x}, \mathbf{z}) = -\log P(\mathbf{x} \mid \mathbf{z}) - \log Z(\mathbf{z}) = \sum_{C \in \mathcal{C}} \Psi_C(\mathbf{x}_C) + \sum_i \Phi_i(x_i, z_i) \tag{3.12}$$

and therefore

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{z}) = \arg\min_{\mathbf{x}} E(\mathbf{x}, \mathbf{z}). \tag{3.13}$$

Thus, the MAP estimate $\hat{\mathbf{x}}$ can be found by minimizing the energy function $E(\mathbf{x}, \mathbf{z})$.

Many problems in computer vision can be posed in terms of energy functions consisting only of unary and pairwise terms (cf. the cliques in the image graph in Figure 3.2(a)). The energy can then be written as

$$E(\mathbf{x}, \mathbf{z}) = \sum_{i \in \mathcal{V}} \Phi_i(x_i, z_i) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i, x_j). \tag{3.14}$$

The first term is referred to as the *data term* and accounts for the likelihood of x_i having observed z_i. The second term is called *prior* or *smoothness term* and incorporates *a priori* knowledge about the model without possessing any further information. Consider for instance the image graph in Figure 3.2(a). Then, Ψ_{ij} may be chosen such that random variables which are neighbors are likely to take the same values.

In the following sections we will discuss several graph cut based algorithms for the exact and approximative minimization of energy functions of the above form.

3.3 Conditional Random Fields

A special form of Markov random fields are the so called *Conditional Random Fields* (CRF), which are used in speech and image analysis. Instead of modeling the posterior as a factorization of the likelihood $P(\mathbf{z} \mid \mathbf{x})$ and the prior $P(\mathbf{x})$ as before, the posterior is directly stated as

$$P(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} e^{-E(\mathbf{x},\mathbf{z})}. \tag{3.15}$$

Note that contrary to (3.10) in this model all terms in E may depend on the observations $\mathbf{z}$. This implicit form allows complex dependencies of $\mathbf{x}$ on $\mathbf{z}$ [10, 80]. Moreover, Li [84] argues that the two main differences between MRFs and CRFs are the following. First, in a CRF the unary potential Φ_i is a function of all $\mathbf{z}$ not just of z_i and of x_i. Second, in a MRF the pairwise potential Ψ_{ij} is independent of the observations whereas in a CRF the potential $\Psi_{ij}(x_i, x_j, \mathbf{z})$ is a function of $\mathbf{z}$ as well as of the realizations x_i and x_j [80].

Another noteworthy detail is that the order of a CRF is defined as the size of the largest clique in contrast to the order of a MRF, where it is defined as the size of the largest clique minus one [57].

CHAPTER 4

Graph Cut Methods for Energy Minimization

In this chapter, we discuss graph cut methods for the exact and approximative inference of the MAP estimate in discrete Markov random fields. In fact, given some observations, for instance a recorded image which has been degraded by noise, the MAP estimate can be derived by minimizing the Gibbs energy as we have seen in Chapter 3. We will focus on graph cut based methods in discrete MRF models where the realizations of the random variables origin from a discrete finite set $\mathcal{L}$ (e.g. the intensity values from a grayscale). In the remainder, $\mathcal{L}$ will be referred to as the *label set*. Furthermore, we speak of a *binary* or *Boolean MRF* if the label set is the set $\mathbb{B}$ and of a *multilabel MRF* if $\mathcal{L}$ contains more than two values.

Let us consider the general form of a first-order MRF energy function

$$E(\mathbf{x}, \mathbf{z}) = \sum_{i \in \mathcal{V}} \Phi_i(x_i, z_i) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i, x_j), \tag{4.1}$$

which can be written as a sum of unary and binary terms.[1] Whilst Φ_i does not affect the tractability, the choice of Ψ_{ij} heavily affects the complexity of the resulting optimization problem. In general, minimizing energy functions of the above form over a set of variables $\mathbf{x}$ with a finite discrete domain $\mathcal{L}$ is NP-hard, even for the binary case (i.e. $\mathcal{L} := \mathbb{B}$) [75], as we will see.

Nevertheless, there exist families of energy functions that can be minimized exactly in polynomial time. An important class are the so called *submodular functions* [45, 87], which are set functions that correspond to energies in (not necessarily Boolean) Markov Random Fields. The minimization of submodular functions has been known to be computable in polynomial time for quite some time [62]. Submodular functions are related to convex continuous functions [86] and, to the best of our knowledge, the fastest algorithm was suggested by Orlin [89] and is a strongly polynomial-time combinatorial algorithm. Unfortunately, for imaging applications, where the number of pixels tends to be huge, such general algorithms are highly impractical.

[1]Note that the variables $\mathbf{z}$ are fixed.

Submodular Boolean Markov Random Field energies can be minimized in polynomial time via graph cuts in an exact manner as shown by Kolmogorov and Zabih [76]. In an early application, Greig et al. [50] inferred the exact maximum a posteriori probability estimate of a degraded black and white image and thereby introduced graph cuts to the field of computer vision. Since then, many practical extensions, for instance to multilabel MRFs, have been made (e.g. see [19, 21, 30, 56, 66]).

The structure of this chapter is as follows. In Sections 4.1 and 4.2 we present pseudo-Boolean and submodular functions since both are directly related to MRF energies. Then, in Section 4.3 we describe the basic idea of using graph cuts for energy minimization. Moreover, in Sections 4.4 and 4.5, we discuss families of MRF energy functions which can be minimized exactly or at least can be approximated with graph cuts. Finally, Sections 4.6 and 4.7 describe the advances in minimizing nonsubmodular energy functions and briefly discuss energy functions with terms that depend on more than two variables.

4.1 Pseudo-Boolean Functions

So far, we have discussed energy functions with a finite discrete labelset $\mathcal{L}$. In this section, we restrict ourselves to functions over variables with a Boolean domain and introduce the so called *Pseudo-Boolean Functions* (PBF):

Definition 10 (Boros and Hammer [11]). *A mapping $f : \mathbb{B}^n \to \mathbb{R}$ is called **pseudo-Boolean function**.*[2]

As with MRF energies we are interested in the optimization problem[3], i.e.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{B}^n} f(\mathbf{x}). \tag{4.2}$$

Let us consider a simple example:

Example 2. *For instance, consider the function $f : \mathbb{B}^2 \to \mathbb{R}$ defined as*

$$f(x_1, x_2) := 4x_1 - 2x_2 + x_1 x_2. \tag{4.3}$$

*f is a **quadratic** pseudo-Boolean function with minimum -2.*

Due to the restriction to the domain $\mathbb{B}$, we can immediately establish a connection to set functions. Let n denote a positive integer and let $V := \{1, \dots, n\}$. Then, it is easy to see that there is a one-to-one correspondence between the elements of the power set of V, denoted by 2^V, and the binary vectors $\mathbf{x} \in \mathbb{B}^n$. A pseudo-Boolean function can thus be considered as a set function mapping a real value to every subset of V. Furthermore, let us denote for a subset $S \subseteq V$ by $\mathbf{x} \in \mathbb{B}^n$ its *characteristic vector*, defined as

$$x_i := \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

[2]The word “pseudo” refers to the fact that pseudo-Boolean functions map to the real numbers.

[3]We treat minimization and maximization equally since $\min f(\mathbf{x}) = -\max -f(\mathbf{x})$.

Obviously, each entry x_i represents a Boolean variable. It is thus convenient to define its *complement* as $\overline{x}_i := 1 - x_i$. Moreover, we call the set $\mathbf{L} = \{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$ the set of *literals*.

The above definitions allow the direct algebraic formulation of many set functions instead of listing an exponential number of values:

Example 3. *Let $V := \{1, 2, \ldots, n\}$ be the ground set and let $f : 2^V \to \mathbb{R}$ be a set function. We define, for every subset $S \subseteq V$, the set function $f(S)$ as the cardinality of S, i.e. $f(S) = |S| = \sum_{1 \leq i \leq n} x_i$.*

As it turns out, pseudo-Boolean functions are quite powerful and many problems in combinatorial optimization and operations research can directly be stated in a pure algebraic manner. Among these are well known problems such as the maximum independent set, vertex cover, and maximum satisfiability problem. It is known that maximization as well as minimization of PBFs is NP-hard in general [11]. Let us illustrate the expressiveness of PBFs with an example:

Example 4. *Consider the following search problem* MAXIMUM INDEPENDENT SET*:*

Instance:	*An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.*
Task:	*Find the largest set $S \subseteq \mathcal{V}$ such that no two vertices in S are adjacent.*

It is known that the problem is NP*-hard [46]. Since the above problem is a set problem, we can easily give a purely algebraic formulation [11]:*

$$\max_{\mathbf{x} \in \mathbb{B}^{\mathcal{V}}} \sum_{i \in \mathcal{V}} x_i - \sum_{(i,j) \in \mathcal{E}} x_i x_j. \tag{4.5}$$

It is useful to define the following:

Definition 11 (Boros and Hammer [11]). *The **multi-linear polynomial representation** of a pseudo-Boolean function is*

$$f(\mathbf{x}) = \sum_{S \subseteq V} c_S \prod_{i \in S} x_i, \tag{4.6}$$

where $c_S \in \mathbb{R}$.

We shall define $\prod_{i \in \emptyset} x_i = 1$. Furthermore, we denote by deg(f) the *degree* of a PBF, which is the cardinality of the largest subset S having $c_S \neq 0$. Consequentially, a pseudo-Boolean function f is said to be *linear* (*quadratic*, *cubic*, etc.) if deg$(f) \leq 1$ (2, 3, etc). This representation is convenient for the study of pseudo-Boolean functions since it is easy to observe the following:

Proposition 1 (Boros and Hammer [11]). *Every pseudo-Boolean function $f : \mathbb{B}^n \to \mathbb{R}$ has a **unique** multilinear polynomial representation of the form (4.6).*

Moreover, Boros and Hammer [11, Sec. 4.4] show that it is possible to reduce any PBF to a unique quadratic representation and give a polynomial-time algorithm. Many reductions with various advantages and disadvantages have been proposed since then [42, 57, 76]. We will discuss this matter in Section 4.7.

It has been known for more than fifty years that there exists a connection between graph cuts and *quadratic pseudo-Boolean functions*, which are of the form

$$f(x_1, \dots, x_n) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{1 \le i < j \le n} c_{ij} x_i x_j. \tag{4.7}$$

Ivănescu [61][4] found that the cut function of a flow network can be written as a quadratic pseudo-Boolean function. He investigated pseudo-Boolean programming to determine the minimum cut of a network and equivalently the maximum flow. Later, Picard and Ratliff [90, 91] established the equivalence and showed that for nonpositive coefficients c_{ij}, a quadratic pseudo-Boolean function such as the above can be minimized efficiently by computing a minimum cut [91, Prpty. 3]. However, they left the problem of minimizing a quadratic pseudo-Boolean function with positive coefficients of the quadratic terms via graph cuts as an open question.

In the next section, we will answer this question and see how this severe restriction to non-positive coefficients relates to the so called *submodular functions* and how this result affects the tractability of minimizing (Boolean) Markov random field energies.

4.2 Submodular Functions

Submodular functions and optimization thereof are a well-studied issues in the field of (combinatorial) optimization with numerous practical applications, for instance in machine learning.[5] First, submodular functions are set functions defined on some finite discrete ground set, and second, such functions fulfill a certain property which make them somehow related to convex (continuous) functions giving an interesting subject for study [86]. In this section, we will see how submodular functions relate to pseudo-Boolean functions and, most important, we will show that the cut function introduced in Section 2.2 is always submodular.

Let V be a finite ground set of cardinality n and let $f : 2^V \to \mathbb{R}$ be a set function assigning each subset of V a real number. Then, we define the following:

Definition 12 (Murota [87])**.** *A set function $f : 2^V \to \mathbb{R}$ is said to be **submodular** if for all sets $X, Y \subseteq V$,*

$$f(X \cup Y) + f(X \cap Y) \le f(X) + f(Y). \tag{4.8}$$

Conversely, a set function f is said to be *supermodular* if $-f$ is submodular and we speak of a *modular* function if f is both submodular and supermodular. Before we continue, let us

[4] A colleague pointed out that Peter L. Ivănescu and Peter L. Hammer are actually the same person according to http://www.gap-system.org/~history/Biographies/Hammer.html.

[5] We refer the reader to http://www.submodularity.org/ for a comprehensive list of topics and applications of submodular functions in machine learning.

consider the following illustrative example from image segmentation which should motivate the study of submodular functions for vision:

Example 5. *Let $X := \{x, y\}$ be the set of pixels of a one-dimensional image and let, for every subset $A \subseteq X$, the function $f(A)$ be the (Gibbs) energy for the pixels in A being classified as foreground object and $X \setminus A$ being classified as background. We shall assume that f is submodular. Then,*

$$f(\{x, y\}) + f(\emptyset) \leq f(\{x\}) + f(\{y\}) \tag{4.9}$$

favors smoothness.

With regard to MRF energies we are interested in the following properties. First, it is easy to see that a function of one variable is always submodular. Second, submodularity is closed under addition and every linear combination of submodular functions f_i with positive coefficients $\alpha_i > 0$, i.e.

$$\sum_i \alpha_i f_i, \tag{4.10}$$

again is submodular.

Lovász [86] points out several examples for submodular functions, many of them arising from graph theory. For instance, he states that the cost function of a cut of a graph is always submodular [86, cf. Example 1.3]. For the sake of completeness, we give a proof.

Claim 1. *The cost function $c : \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ of every s-t cut in every flow network $(\mathcal{G}, c)$ is submodular.*

For convenience let us define a modified cut function $\hat{c} : 2^{\mathcal{V}} \to \mathbb{R}^+$ as

$$\hat{c}(S) := c(S, \mathcal{V} \setminus S).$$

Proof. Let $S_1, S_2 \subset \mathcal{V}$ be two arbitrary cuts on $\mathcal{G}$. For the submodularity of $\hat{c}$, we have to show that

$$\hat{c}(S_1 \cup S_2) + \hat{c}(S_1 \cap S_2) \leq \hat{c}(S_1) + \hat{c}(S_2).$$

First, observe that either of S_1 and S_2 can be decomposed into two disjoint sets such that $S_1 = (S_1 \setminus S_2) \cup (S_1 \cap S_2)$ and $S_2 = (S_2 \setminus S_1) \cup (S_1 \cap S_2)$. The same holds for the complement, i.e. $\overline{S_1} = (\overline{S_1} \setminus S_2) \cup (S_2 \setminus S_1)$ and $\overline{S_2} = (\overline{S_2} \setminus S_1) \cup (S_1 \setminus S_2)$. In further consequence, we can write the cost of the cuts as

$$\begin{aligned}
\hat{c}(S_1) &= c(S_1 \setminus S_2, \overline{S_1}) + c(S_1 \cap S_2, \overline{S_1}) \\
&= c(S_1 \setminus S_2, \overline{S_1} \setminus S_2) + c(S_1 \setminus S_2, S_2 \setminus S_1) + c(S_1 \cap S_2, \overline{S_1} \setminus S_2) + c(S_1 \cap S_2, S_2 \setminus S_1) \\
\hat{c}(S_2) &= c(S_2 \setminus S_1, \overline{S_2}) + c(S_1 \cap S_2, \overline{S_2}) \\
&= c(S_2 \setminus S_1, \overline{S_2} \setminus S_1) + c(S_2 \setminus S_1, S_1 \setminus S_2) + c(S_1 \cap S_2, \overline{S_2} \setminus S_1) + c(S_1 \cap S_2, S_1 \setminus S_2)
\end{aligned}$$

and

$$\begin{aligned}\hat{c}(S_1 \cup S_2) &= c(S_1 \setminus S_2, \overline{S_1} \setminus S_2) + c(S_2 \setminus S_1, \overline{S_2} \setminus S_1) + c(S_1 \cap S_2, \overline{S_1} \setminus S_2)\\ \hat{c}(S_1 \cap S_2) &= c(S_1 \cap S_2, S_1 \setminus S_2) + c(S_1 \cap S_2, S_2 \setminus S_1) + c(S_1 \cap S_2, \overline{S_1} \setminus S_2)\end{aligned}$$

It is now easy to see that by substituting the terms in

$$\hat{c}(S_1 \cup S_2) + \hat{c}(S_1 \cap S_2) \leq \hat{c}(S_1) + \hat{c}(S_2)$$

we have

$$0 \leq c(S_1 \setminus S_2, S_2 \setminus S_1) + c(S_2 \setminus S_1, S_1 \setminus S_2) = \sum_{\substack{(u,v)\in\mathcal{E}\\ u\in S_1\setminus S_2\\ v\in S_2\setminus S_1}} c(u,v) + \sum_{\substack{(u,v)\in\mathcal{E}\\ u\in S_2\setminus S_1\\ v\in S_1\setminus S_2}} c(u,v).$$

Since the capacities of the arcs are always nonnegative and since S_1 and S_2 were arbitrary cuts the claim follows.[6] □

In Section 4.1 we stated the direct connection between set functions and pseudo-Boolean functions via characteristic vectors. It is thus possible to define submodularity on PBFs. A pseudo-Boolean function f is submodular if and only if

$$f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathbb{B}^n, \tag{4.11}$$

where $\mathbf{x} \vee \mathbf{y}$ and $\mathbf{x} \wedge \mathbf{y}$ denote the vectors of componentwise maxima and minima, respectively, defined as

$$(\mathbf{x} \vee \mathbf{y})_i := \max\{x_i, y_i\} \text{ and } (\mathbf{x} \wedge \mathbf{y})_i := \min\{x_i, y_i\}. \tag{4.12}$$

Example 6. *Let $\Psi : \mathbb{B}^2 \to \mathbb{R}$ be an arbitrary term arising from a Boolean MRF energy. Then, Ψ is submodular if and only if*

$$\Psi(0,0) + \Psi(1,1) \leq \Psi(0,1) + \Psi(1,0). \tag{4.13}$$

Regarding the submodularity of pseudo-Boolean functions, the following necessary and sufficient condition can be stated:[7]

Proposition 2. *A quadratic pseudo-Boolean function of the form*

$$f(\mathbf{x}) = \sum_{1\leq i,j\leq n} c_{ij} x_i x_j + \sum_{1\leq i\leq n} c_i x_i + c_0 \tag{4.14}$$

is submodular if and only if $c_{ij} \leq 0$ for all i, j.

[6]The submodularity of the cut function of an undirected weighted graph can be shown in a similar way.
[7]Follows as a direct consequence from Nemhauser et al. [88, Prop. 3.5].

Let us develop the above proposition by a simple example:

Example 7. *The function $\Psi : \mathbb{B}^2 \to \mathbb{R}$ from Example (6) can equivalently be stated as the following PBF:*

$$f(x_1, x_2) = \Psi(0,0)\overline{x}_1\overline{x}_2 + \Psi(1,1)x_1x_2 + \Psi(0,1)\overline{x}_1x_2 + \Psi(1,0)x_1\overline{x}_2 \tag{4.15}$$

and by substituting negated variables $\overline{x}_i$ by $1 - x_i$ transformed to

$$f(x_1, x_2) = \big(\Psi(0,0) + \Psi(1,1) - \Psi(0,1) - \Psi(1,0)\big)x_1x_2 + L, \tag{4.16}$$

where L are all positive linear terms (see Freedman and Drineas [42, Sec. 2.1] for the details of the reduction).

The immediate consequence of this equivalence is the following theorem:

Theorem 3 (Freedman and Drineas [42, Thm. 1])**.** *A quadratic pseudo-Boolean function can be minimized via graph cut techniques if and only if it is submodular.*

Clearly, by Theorem 3 and Proposition 2, the pseudo-Boolean function f from Example 7 can be minimized exactly with graph cuts if and only if equation (4.13) holds.

An alternative definition of submodularity is given by Schrijver [106] and is equivalent to Definition (12):

Theorem 4 (Schrijver [106, Thm. 44.1])**.** *A set function f on a finite ground set V is submodular if and only if*

$$f(X) + f(X \cup \{i,j\}) \leq f(X \cup \{i\}) + f(X \cup \{j\}) \tag{4.17}$$

for each $X \subseteq V$ and distinct $i, j \in V \setminus X$.

Even though by equation (4.11) submodularity is instantly defined for pseudo-Boolean functions of arbitrary order, the theorem above provides a convenient way of checking submodularity of f via a single vector $\mathbf{x} \in \mathbb{R}^n$: fix all values but two and then check whether submodularity holds. For the sake of completeness, it shall be mentioned that this idea was independently developed by Kolmogorov [71], and Kolmogorov and Zabih [76] who referred to it as *regularity*. The equivalence between these concepts is shown by Freedman and Drineas [42].

The notion of submodularity generalizes to functions defined on $\mathbb{R}^n$ such that a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be submodular if and only if

$$f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{4.18}$$

holds [87, Eq. 2.17]. As a consequence, submodularity is defined for every linearly ordered set $\mathcal{L}$ [30, Eq. 2]. It is noteworthy that the submodularity of a function depends on the order of the set.

We refer the reader to the work of Fujishige [45], Murota [87], and Schrijver [106] for further details on the subject of submodular functions.

4.3 Graph Cuts for Boolean MRF Energies

In Section 4.1, we have stated the direct correspondence between submodular quadratic pseudo-Boolean functions and minimum cuts. In this section, we will see how to represent Boolean Markov random field energies with submodular priors as quadratic pseudo-Boolean functions and show how to construct a graph such that the minimum cut globally minimizes the energy.

Recall the image graphs from Section 3.2 and consider an *undirected* graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ arising from a regular grid in its general form, i.e. each pixel corresponds to a vertex in the graph (cf. Figure 3.2). We will now construct a new graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ which we will use as a flow network to encode the Boolean energy.

First, let us define the new set of nodes as $\hat{\mathcal{V}} := \mathcal{V} \cup \{s, t\}$, i.e. we add a source and a sink. The set $\hat{\mathcal{E}}$ consists of two types of edges: *t-links* (terminal links) and *n-links* (neighborhood links). The t-links connect each node to both terminals whereas the n-links connect neighboring pairs as in the original graph. Thus, $\hat{\mathcal{E}} := \mathcal{E} \bigcup_{v \in \mathcal{V}} \{(s, v), (v, t)\}$. A subtle difference emerges from the fact that $\mathcal{G}$ was undirected whereas we have defined $\hat{\mathcal{G}}$ is a directed graph. For the time being let us assume that for every undirected edge $\{i, j\} \in \mathcal{E}$ we add both arcs (i, j) and (j, i) to $\hat{\mathcal{V}}$. For better understanding the constructed graph is illustrated in Figure 4.1 (image taken from Boykov and Kolmogorov [18]).

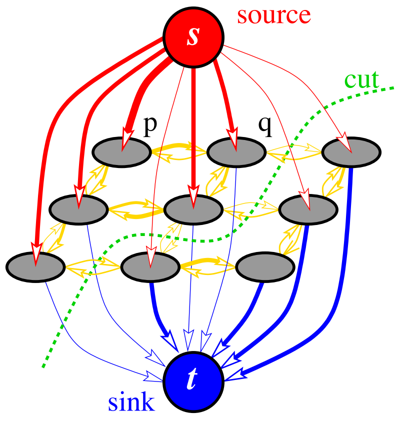


Figure 4.1: The constructed flow network.

Still, our graph lacks the arc capacities to graduate to a full flow network. Before dealing with them let us analyze the constructed network. Figure 4.3 shows the final network for a one-dimensional grid with two pixels $\{x_i, x_j\}$. The dashed lines indicate all possible cuts in the depicted network.

The crucial insight is now that the flow network is constructed in a way such that each s-t cut corresponds to a configuration $\mathbf{x} \in \mathbb{B}^n$, since for any minimum s-t cut (S, T) we can define:

$$x_i := \begin{cases} 0 & \text{if } i \in S, \\ 1 & \text{if } i \in T, \end{cases} \tag{4.19}$$

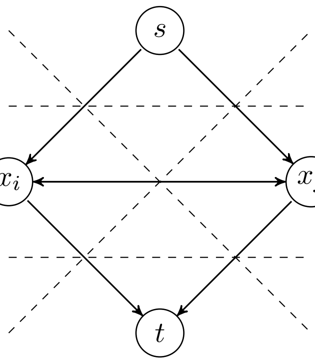


Figure 4.2: Constructed network.

for all $i \in \mathcal{V}$. Thus, all variables corresponding to vertices being in the same partition as the source get assigned 0 and all the others get assigned 1. Let us show that this assignment is well defined. Therefore, let $\mathcal{J}$ be the set of feasible cuts on $\hat{\mathcal{G}}$ which satisfy the following two properties:

1. for each node $i \in \hat{\mathcal{V}} \setminus \{s,t\}$, exactly one of (s,i) and (i,t) is cut, and
2. each n-link (i,j) is cut if and only if $i \in S$ and $j \in T$.

Yet, we have to show that these properties hold for a minimum s-t cut.

Claim 2. *A minimum cut $\hat{C}$ on $\hat{\mathcal{G}}$ is feasible, i.e. $\hat{C} \in \mathcal{J}$.*

Proof. Taken from [13]. Since $\hat{C}$ separates the two terminals, exactly one of (s,i) and (i,t) is in $\hat{C}$. On the other hand, if both edges are in the cut, $\hat{C}$ would not be a minimal. Thus, property 1) holds. Property 2) follows since if $i \in S$ and $j \in T$ but $(i,j) \notin \hat{C}$, there would be an s-t path. If i and j are in the same partition of the cut, then $(i,j) \notin \hat{C}$ because of the minimality of $\hat{C}$. $\square$

Now that we have established the relation between s-t cuts and configurations, we may write the cost $c(S,T)$ of a cut as

$$c(S,T) = \sum_{i \in \mathcal{V}} (c(s,i)x_i + c(i,t)\overline{x}_i) + \sum_{(i,j)\in\hat{\mathcal{E}}} (c(i,j)\overline{x}_i x_j + c(j,i)x_i\overline{x}_j)\,. \tag{4.20}$$

From that it follows immediately that by finding a minimum cut on $\hat{\mathcal{G}}$ one minimizes the above quadratic pseudo-Boolean function.

In the remainder we will use a notation introduced by Kolmogorov and Rother [72], which captures the values of the above function as follows: θ_{const} is a constant term not reflected by the cut. For each $i \in \mathcal{V}$, we define θ_{i,x_i} as $\theta_{i;x_i} := c(s,i)x_i + c(i,t)\overline{x}_i$, and for each $(i,j) \in \hat{\mathcal{E}}$ we define $\theta_{ij:x_ix_j}$ as $\theta_{ij:x_ix_j} := c(i,j)\overline{x}_i x_j + c(j,i)x_i\overline{x}_j$. Thus, all information can be concatenated to a tensor θ which also contains the constant θ_{const}.

Based on the construction of the above graph and the restrictions that all edge capacities are nonnegative, we are safe to state that

$$\theta_{i;0} \geq 0 \qquad \theta_{i;1} \geq 0 \tag{4.21}$$

$$\theta_{ij;00} = 0 \qquad \theta_{ij;01} \geq 0 \qquad \theta_{ij;10} \geq 0 \qquad \theta_{ij;11} = 0 \tag{4.22}$$

holds for all nodes $i \in \mathcal{V}$ and all arcs $(i,j) \in \hat{\mathcal{E}}$. The first and the last statement in (4.22) impose a sever restriction on the pseudo-Boolean functions we can minimize with the above construction: the class of submodular quadratic pseudo-Boolean functions. The question now is how to generalize this observation to 1) arbitrary submodular quadratic PBFs having $\theta_{ij;00}, \theta_{ij;11} > 0$, and 2) arbitrary (submodular) energies. For the time being we will consider the quadratic case only, since in Section 4.1 we mentioned that every PBF of degree higher than two can be reduced to a unique quadratic PBF.

Kolmogorov and Rother [72] introduced the concept of *reparameterization* of a Boolean MRF energy to computer vision, which has been familiar to the machine learning community for some time. Given some pseudo-Boolean energy, the idea is to apply a (finite) number of steps which rewrite the terms in the energy function such that the constant term increases while the binary terms at some point comply with (4.22). The reparameterization process is as follows.

Let us start off with an unary term $\theta_{i:a}$. Given some node $i \in \mathcal{V}$ and some $\delta \in \mathbb{R}^+$, we transform θ such that

$$\theta_{i;0} \leftarrow \theta_{i;0} - \delta \qquad \theta_{i;1} \leftarrow \theta_{i;1} - \delta \qquad \theta_{const} \leftarrow \theta_{const} + \delta, \tag{4.23}$$

where $\leftarrow$ denotes the assignment operator as used with pseudocode. The crucial observation is that the transformation preserves the energy $E(\mathbf{x})$, since $x_i + \overline{x}_i = 1$. Informally speaking, each reparameteriztion step "shaves off" an additive constant.

Considering the identities $\overline{x}_j = (x_i + \overline{x}_i)\overline{x}_j$ and $x_j = (x_i + \overline{x}_i)x_j$ for some arc $(i,j) \in \hat{\mathcal{E}}$ motivates the reparameterization operations

$$\theta_{ij;00} \leftarrow \theta_{ij;00} - \delta \qquad \theta_{ij;10} \leftarrow \theta_{ij;10} - \delta \qquad \theta_{j;0} \leftarrow \theta_{j;0} + \delta \tag{4.24}$$

and

$$\theta_{ij;01} \leftarrow \theta_{ij;01} - \delta \qquad \theta_{ij;11} \leftarrow \theta_{ij;11} - \delta \qquad \theta_{j;1} \leftarrow \theta_{j;1} + \delta \tag{4.25}$$

The same holds true for the identities $\overline{x}_i = \overline{x}_i(x_j + \overline{x}_j)$ and $x_i = x_i(x_j + \overline{x}_j)$, which we don't state for brevity reasons.

The crucial observation is that for submodular vectors $\boldsymbol{\theta}$ the transformation preserves submodularity. Clearly, modifying unary energies $\theta_{i;a}$ does not affect submodularity. Let us consider a pairwise energy $\theta_{ij;ab}$ for which submodularity is defined as (cf. Section 4.2)

$$\theta_{ij;00} + \theta_{ij;11} \leq \theta_{ij;01} + \theta_{ij;10}. \tag{4.26}$$

It is easy to see that none of the described operations violates submodularity. Having gained this insight, we are ready to generalize the functions minimized by the above cut function.

Definition 13 (Kolmogorov and Rother [72])**.** *If two parameter vectors θ and θ' define the same Boolean MRF energy, i.e. $E(\mathbf{x}, \theta) = E(\mathbf{x}, \theta')$ for all configurations $\mathbf{x} \in \mathbb{B}^n$, then θ' is called a **reparameterization** of θ and is denoted by $\theta' \sim \theta$.*

The reparameterization motivates the following definition:[8]

Definition 14 (Blake et al. [10])**.** *A parameter vector θ is said to be in **normal form** if for each node $i \in \mathcal{V}$*

$$\min\{\theta_{i;0}, \theta_{i;1}\} = 0 \tag{4.27}$$

holds and for each arc $(i, j) \in \hat{\mathcal{E}}$ one of (4.28) and (4.29) is satisfied:

$$\theta_{ij;00} = 0 \qquad \theta_{ij;01} \geq 0 \qquad \theta_{ij;10} \geq 0 \qquad \theta_{ij;11} = 0 \tag{4.28}$$

$$\theta_{ij;00} \geq 0 \qquad \theta_{ij;01} = 0 \qquad \theta_{ij;10} = 0 \qquad \theta_{ij;11} \geq 0. \tag{4.29}$$

Based on the above observations, Blake et al. [10, Sec. 2.2.1] give a simple algorithm for computing the normal form of a vector θ:

```
1. For each arc $(i,j) \in \hat{\mathcal{E}}$:
    a) Compute $\delta \leftarrow \min_{a,b\in\mathbb{B}} \theta_{ij:ab}$ and update
        i. $\theta_{ij;ab} \leftarrow \theta_{ij;ab} - \delta, \quad \forall a, b \in \{0, 1\}$,
        ii. $\theta_{const} \leftarrow \theta_{const} + \delta$.
    b) For each $b \in \{0, 1\}$ compute $\delta \leftarrow \min\{\theta_{ij;0b}, \theta_{ij;1b}\}$ and update
        i. $\theta_{ij;0b} \leftarrow \theta_{ij;0b} - \delta$; $\theta_{ij;1b} \leftarrow \theta_{ij;1b} - \delta$; $\theta_{j;b} \leftarrow \theta_{j;b} + \delta$.
    c) For each $a \in \{0, 1\}$ compute $\delta \leftarrow \min\{\theta_{ij;a0}, \theta_{ij;a1}\}$ and update
        i. $\theta_{ij;a0} \leftarrow \theta_{ij;a0} - \delta$; $\theta_{ij;a1} \leftarrow \theta_{ij;a1} - \delta$; $\theta_{i;a} \leftarrow \theta_{i;a} + \delta$.
2. For each node $i \in \mathcal{V}$ compute $\delta \leftarrow \min\{\theta_{i;0}, \theta_{i;1}\}$ and update
    a) $\theta_{i;0} \leftarrow \theta_{i;0} - \delta$; $\theta_{i;1} \leftarrow \theta_{i;1} - \delta$; $\theta_{const} \leftarrow \theta_{const} + \delta$.
```

The runtime of the algorithm is $O(|\mathcal{V}| + |\mathcal{E}|)$, since each iteration computes a constant number of steps for both loops.

It shall be noted that the normal form is not unique in general and moreover, Blake et al. [10, Sec. 2.5.1] argue that the reparameterization relates to the problem of determining the largest lower bound, i.e. $\theta_{const} \leq \min_{\mathbf{x}} E(\mathbf{x}, \theta)$, on the reparameterized energy. This problem again can be solved by a maximum flow computation.

At this point, we are ready to tie everything together. The following theorem follows from the established facts:

Theorem 5 (Blake et al. [10])**.** *For any submodular quadratic pseudo-Boolean function, the minima $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} E(\mathbf{x}, \theta)$ can be obtained in polynomial time by the following steps:*

[8] The original idea was developed in Kolmogorov and Rother [72]. Nevertheless, the definition used by Blake et al. [10] is easier to grasp.

1. *Reparameterize θ to a normal form with the presented algorithm.*
2. *Construct a flow network $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ as described with the following nonnegative capacities:*

$$c(s,i) := \theta_{i;1} \qquad c(i,t) := \theta_{i;0} \qquad c(i,j) := \theta_{ij;01} \qquad c(j,i) := \theta_{ji;10}. \tag{4.30}$$

3. *Compute a minimum s-t cut in $\hat{\mathcal{G}}$ and determine* $\mathbf{x}$.

Computing a minimum s-t cut can be done by any of the (polynomial-time) maximum flow algorithms mentioned in Section 2.2. For the the second part, i.e. determining x_i for each $i \in \mathcal{V}$, we use the fact that an s-t cut (S, T) severs the terminals s and t. Thus, S can be determined by finding all nodes reachable from s (cf. Kleinberg and Tardos [67]). For simple flow networks such as ours it is sufficient to check for each $i \in \mathcal{V}$ wether arc (s, i) is saturated by the determined maximum flow and thus part of the minimum cut.

We have shown how to efficiently use graph cuts for energy minimization of submodular pseudo-Boolean MRF energies. Unfortunately, for nonsubmodular energies the presented approach can not be applied. As a matter of fact, nonsubmodular pseudo-Boolean function minimization in general is NP-hard. In Section 4.6 we will justify this claim by a proof. Nevertheless, we will discuss an approach which allows at least the computation of a partial solution.

4.4 Exact Minimization of Multilabel MRF Energies

In Section 4.3, we have seen the graph construction for the exact minimization of Boolean MRF energies. Nevertheless, most (interesting) applications such as denoising require a multilabel setting, i.e. $|\mathcal{L}| > 2$. In this section, we cover the basic ideas of graph constructions which allow the exact minimization of MRF energies over n variables from the domain $\mathcal{L}$ and discuss the restrictions imposed on these functions.

It shall be mentioned that Hochbaum [54] developed a polynomial-time algorithm for energies with a convex unary potentials $\Phi(x_i, z_i)$ and linear priors $\Psi(x_i, x_j)$, and a strongly polynomial-time algorithm when the unary potentials are linear, quadratic or piecewise linear convex with "few" pieces (see Hochbaum [54] for the details), respectively. Moreover, Charpiat [25] recently suggested a graph construction which adds some nonsubmodular energies to the set of functions that can be minimized exactly with graph cuts.

Convex Priors

Ishikawa [56] suggested a graph construction which is able to minimize MRF energies with *convex* priors.[9] A function $g(\delta)$ defined on a set of consecutive integers, i.e. $\mathcal{L} := \{1, \ldots, k\}$, is convex if and only if all second differences are nonnegativ, i.e. $g(\delta+1) - 2g(\delta) + g(\delta-1) \geq 0$ holds. Let us look at some examples which are frequently used in computer vision.

Example 8. *Let $\delta := x_i - x_j$. A linear (and increasing) convex function is $\Psi(x_i, x_j) := |\delta|$ and a quadratic convex function is $\Psi(x_i, x_j) := |\delta|^2$.*

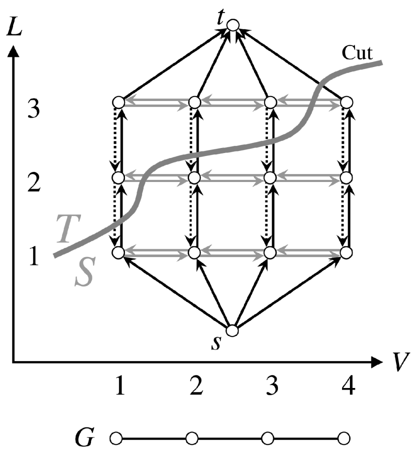


Figure 4.3: The construction for convex priors.

Having discussed the prerequisites, let us look at the graph construction. Given an image graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a labelset $\mathcal{L} := \{1, \ldots, k\}$, the suggested construction introduces $k - 1$ nodes $u_{i,l}$, one for each $i \in \mathcal{V}$ and every label $1 \leq l < k$, representing the elements of $\mathcal{L}$. The set of arcs consists of three types: *data edges*, which for each i form a path from s to t via the introduced label nodes and account for Φ_i, *constraint edges*, which are of infinite capacity and form the reverse path for each $i \in \mathcal{V}$ to prevent a cut from "going backwards", and finally the *penalty edges*, which connect each label node to all the other label nodes of neighboring nodes (pixels) in the original graph and account for the pairwise potential Ψ_{ij}. The construction for a simple one-dimensional graph and three labels is depicted in Figure 4.3 (taken from Ishikawa [56]). We refer the reader to Ishikawa [56] for the full details of the construction and more illustrative material. Nevertheless, we want to emphasize that any possible s-t cut must sever each path from s to t formed by the introduced nodes $u_{i,l}$ and thus assigns each node a label.

Finally, it shall be noted that the constructed graph consists of $O(|\mathcal{V}| \cdot |\mathcal{L}|)$ nodes and the number of edges depends on the chosen neighborhood. Moreover, in case the set of labels is real-valued and the elements can be indexed by integers (as long as an order is useful for the respective application) the construction can be applied.

Submodular Priors

To the best of our knowledge, Schlesinger and Flach [104] were the first to give a graph construction that minimizes general submodular energies. Recently, Darbon [30] developed a formulation trough *level sets* which allows the exact minimization of MRF energies with submodular priors. Informally, a level set $\lambda \in \mathcal{L}$ consists of all variables having assigned the same label. The basic idea is to introduce for each variable and each label a new binary variable which states whether the value of the variable is less (or equal) or greater than that label.[10]

[9] The construction was already published earlier in Ishikawa and Geiger [60].

[10] Note that by the submodularity $\mathcal{L}$ is totally ordered.

Given an image graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the number of nodes in the constructed graphs is $\Theta(|\mathcal{V}| \cdot |\mathcal{L}|)$ (one for each node and label) and the number of edges is $O(|\mathcal{E}| \cdot |\mathcal{L}|^2)$. It is noteworthy that both the constructions of Darbon [30] and Ishikawa [56] (see previous section) construct a graph which depends on the size of $\mathcal{L}$ and thus the runtime is pseudo-polynomial in the size of the input.

Darbon [30] lists several examples for submodular functions which can be globally minimized with this construction. A widely used submodular prior for instance is $\Psi(x_i, x_j) := g(x_i - x_j)$, where g is an unary convex function. We will use this fact later in our analysis.

4.5 Move-Making Algorithms

In Section 4.3, we presented solutions to minimize Boolean MRF energies and in particular we discussed the exact minimization of submodular Boolean MRF energies. However, a great variety of problems in vision demand more than two labels. For instance, consider image denoising of grayscale images where the values taken by the random variables in the MRF originate from a set $\mathcal{L}$. In general $\mathcal{L}$ may be continuous (in Section 4.6 we discuss this matter). Nevertheless, we will focus on problems for which $\mathcal{L}$ is discrete and the elements can be indexed by k consecutive integers. As before, we state the general form of a first-order MRF energy as

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \Phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i, x_j). \tag{4.31}$$

The major difference to the Boolean case is that in the multilabel scenario $\mathbf{x}$ is chosen from $\mathcal{L}^n$. Even for a small k, it is highly impracticable to minimize MRF energies for images of realistic size by enumeration of all possible labelings.

A possible choice for the priors in (4.31) is the so called *Generalized Potts model*, which was introduced by Boykov et al. [19]. As the name suggests it is a generalization of the original *Potts model* [92] and is defined as

$$\Psi_{ij}(\alpha, \beta) := K_{ij} \cdot T(\alpha \neq \beta), \tag{4.32}$$

where $T(\cdot)$ is 1 if the argument validates to true and 0 otherwise. K_{ij} denotes the cost at the boundaries and thus the model favors *piecewise constant* solutions. In case all K_{ij} are equal, we obtain the original Potts model. By reduction from the minimum cost multiway cut problem it can be shown that minimizing the Potts energy $E(\mathbf{x})$ with priors (4.32) is NP-hard [22, Appx.][11] making it necessary tackle the problem with approximations.

A possible way to deal with this result are so called *move-making* algorithms. These algorithms are characterized by the fact that instead of changing the value of a single variable x_i at a time, which is often referred to as a *standard move* and used both in *Simulated Annealing* [47] and *Iterated Conditional Models* [7] resulting in an exponential worst case runtime, they allow the change of multiple variables at each iteration.

To this end, in each iteration a potential new solution, also referred to as a *move*, is suggested. It is then possible to state this decision procedure as a Boolean energy function which decides

[11]The NP-hardness result holds even for planar MRF image graphs. See Veksler [111, Appx.] for a proof sketch.

for each variable whether it keeps its old value or switches to the suggested. In case the resulting Boolean energy is submodular, the optimal move, i.e. the move which results in the maximum decrease in energy, can be found efficiently with a graph cut as seen in Section 4.3. In case it is nonsubmodular, we will still be able to apply a method called *BHS Algorithm* as we will see later in Section 4.6. Move-making algorithms iterate and propose new solutions until no move can further improve the energy of the current solution. Unfortunately, this "steepest descent" technique may terminate in a local optimum. At least for one move-making algorithm, the expansion algorithm, we will be able to state an upper bound on the quality of the local (energy) minimum.

Two major algorithms have been proposed: the *Expansion Algorithm* and the *Swap Algorithm*. Both were originally developed by Veksler [111] and Boykov et al. [19, 20, 21, 22] and impose different restrictions on the MRF energies to be applicable. In particular, the restriction is on the priors Ψ_{ij}, since Φ_i does not affect the complexity. For the applicability of the expansion algorithm we require all priors Ψ_{ij} to satisfy

$$\Psi_{ij}(\alpha,\alpha) + \Psi_{ij}(\beta,\gamma) \leq \Psi_{ij}(\beta,\alpha) + \Psi_{ij}(\alpha,\gamma), \quad \forall \alpha,\beta,\gamma \in \mathcal{L}, \tag{4.33}$$

whereas the swap algorithm is less restrictive and demands

$$\Psi_{ij}(\alpha,\alpha) + \Psi_{ij}(\beta,\beta) \leq \Psi_{ij}(\alpha,\beta) + \Psi_{ij}(\beta,\alpha), \quad \forall \alpha,\beta \in \mathcal{L} \tag{4.34}$$

to hold for all priors.

The above inequalities arise from the requirement that the Boolean energy of determining the optimal move should be submodular. In the following sections we will explain in detail how they arise.

Special attention deserve the so called *discontinuity-preserving* priors. Discontinuities, for instance object boundaries, in an image should be preserved and not smoothed radically. In other terms, the penalty accounted for boundaries should be bound, e.g. by some constant K. Such priors are of great importance for many problems such as image segmentation and denoising. Unfortunately, even the simplest discontinuity-preserving prior renders the problem NP-hard [22].

In the following we present the two major move-making algorithms, the expansion and the swap algorithm. We discuss how to determine the optimal move from an exponential number of moves and approximation guarantees. Therefore, let us define the underlying moves of the algorithms.

Given a labeling $\mathbf{x}$, we say that a move from $\mathbf{x}$ to $\mathbf{x}'$ is an *expansion* if for all $i \in \mathcal{V}$, $x_i \neq x'_i$ implies $x'_i = \alpha$ for some label $\alpha \in \mathcal{L}$. In other words, variables may change their label to α. Thus, such a move is also referred to as an α*-expansion*.

Given a labeling $\mathbf{x}$, a move from $\mathbf{x}$ to $\mathbf{x}'$ is said to be a *swap* if for all $i \in \mathcal{V}$, $x_i \neq x'_i$ implies $x_i, x'_i \in \{\alpha,\beta\}$ for some pair of labels $\alpha,\beta \in \mathcal{L}$. In other words, a variable may swap its value from α to β or vice versa. This change is often referred to as an α-β-swap.

Both algorithms are similar in their structure. Algorithm 4.1 depicts the expansion algorithm and Algorithm 4.2 the swap algorithm. Starting from an (arbitrary) initial labeling, in each cycle (lines 3-9) the optimal move is determined (line 4). The move is immediately accepted in case the energy decreases. The algorithms iterate until convergence w.r.t. α-expansions and α-β-swaps, respectively. It is readily seen that the expansion algorithms performs $|\mathcal{L}|$ cycles whereas the swap algorithm cycles $|\mathcal{L}|^2$ times in each iteration.

```
   Input: initial labeling $\mathbf{x}$.
   Result: local minimum $\mathbf{x}$.
1  repeat
2      changed $\leftarrow$ false;
3      for each label $\alpha \in \mathcal{L}$ do
4          $\tilde{\mathbf{x}} \leftarrow \arg\min E(\tilde{\mathbf{x}})$ among $\tilde{\mathbf{x}}$ within one $\alpha$-expansion of $\mathbf{x}$;
5          if $E(\tilde{\mathbf{x}}) < E(\mathbf{x})$ then
6              $\mathbf{x} \leftarrow \tilde{\mathbf{x}}$;
7              changed $\leftarrow$ true;
8          end
9      end
10 until changed = false;
```

Algorithm 4.1: Pseudocode for the expansion algorithm.

```
   Input: initial labeling $\mathbf{x}$.
   Result: local minimum $\mathbf{x}$.
1  repeat
2      changed $\leftarrow$ false;
3      for each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$ do
4          $\tilde{\mathbf{x}} \leftarrow \arg\min E(\tilde{\mathbf{x}})$ among $\tilde{\mathbf{x}}$ within one $\alpha$-$\beta$-swap of $\mathbf{x}$;
5          if $E(\tilde{\mathbf{x}}) < E(\mathbf{x})$ then
6              $\mathbf{x} \leftarrow \tilde{\mathbf{x}}$;
7              changed $\leftarrow$ true;
8          end
9      end
10 until changed = false;
```

Algorithm 4.2: Pseudocode for the swap algorithm.

The crucial idea is to state the problem of finding the optimal move in line 4 for a given configuration $\mathbf{x}$ as a Boolean energy minimization problem. Each variable x_i can either keep its old label or switch. Thus, in case the resulting Boolean energy is submodular it can be minimized exactly with a graph cut (cf. Sections 4.2 and 4.3). Nevertheless, one should be aware of the fact that the number of possible moves grows exponentially with $|\mathcal{V}|$.

In order to state the problem of finding the optimal move, let us formally develop the above described. Therefore, let $\mathbf{y} := (y_i)_{i \in \mathcal{V}} \in \mathbb{B}^n$ denote a Boolean vector and let $\mathbf{x}^c$ be the *transformation function*. Given a current configuration $\mathbf{x}^0$ and a move $\mathbf{y}$, the function $\mathbf{x}^c$ computes the new labeling induced by $\mathbf{y}$ as

$$\mathbf{x}^c(\mathbf{y}) := \mathbf{x}^0 \circ (1 - \mathbf{y}) + \mathbf{x}^1 \circ \mathbf{y}, \qquad (4.35)$$

where $A \circ B$ is the Hadamard (elementwise) product defined as $(A \circ B)_i = (A)_i \cdot (B)_i$ and $\mathbf{x}^1$ is a proposed labeling. For the expansion algorithm the proposal is defined as $\mathbf{x}^1 := (\alpha)_i$ for some label $\alpha \in \mathcal{L}$ whereas for the swap algorithm we define $\mathbf{x}^0$ constant as α and $\mathbf{x}^1$ constant as β.

From the above definitions it becomes clear that the standard moves, the expansion, and the swap moves are special cases of the quite general transformation function (4.35). Indeed, Lempitsky et al. [81] were the first to state this general form, which is referred to as the *fusion move*. In Section 4.6, we will see how to deal with such general moves. For the time being we will focus on expansion and swap moves, since the restriction imposed by inequalities (4.33) and (4.34) allow the exact minimization of the resulting Boolean energy with graph cuts.

It is now convenient to denote the MRF energy of the new labeling as $E(\mathbf{x}^c(\mathbf{y}))$. Recall that the problem of finding the optimal move is then

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} E(\mathbf{x}^c(\mathbf{y})). \tag{4.36}$$

Note that the minimization is over the Boolean vector $\mathbf{y}$. The energy can thus be stated naturally as

$$E(\mathbf{x}^c(\mathbf{y})) = \sum_{i \in \mathcal{V}} \Phi_i(x_i^c(y_i)) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i^c(y_i), x_j^c(y_j)). \tag{4.37}$$

As we have seen in Section 4.2, the Boolean energy (4.37) can be minimized exactly with graph cuts if and only if all Ψ_{ij} are submodular. We should keep in mind that for the expansion move the minimum s-t cut decides for each variable x_i^c whether it keeps its label, i.e. x_i^0 if $y_i = 0$, or switches, i.e. $x_i^0 = \alpha$ if $y_i = 1$. For the expansion move, Ψ_{ij} is submodular if for all $\alpha \in \mathcal{L}$ the inequality

$$\Psi_{ij}(\alpha, \alpha) + \Psi_{ij}(x_i^0, x_j^0) \leq \Psi_{ij}(x_i^0, \alpha) + \Psi_{ij}(\alpha, x_j^0) \tag{4.38}$$

is satisfied.[12] Consequentially, for the swap move the energy is submodular if and only if for all labels $\alpha, \beta \in \mathcal{L}$ the inequality

$$\Psi_{ij}(\alpha, \alpha) + \Psi_{ij}(\beta, \beta) \leq \Psi_{ij}(\alpha, \beta) + \Psi_{ij}(\beta, \alpha) \tag{4.39}$$

is satisfied. From that we have established the conditions (4.33) and (4.34) from the beginning of the section. Constructing the corresponding flow network for energy (4.37) is straightforward as discussed in Section 4.3.

For historical reasons, it is noteworthy that in the original work of Veksler [111] and Boykov et al. [22] it was assumed that for the applicability of the expansion and the swap algorithm a prior must be a metric and a semi-metric on $\mathcal{L}$, respectively. However, the work of Kolmogorov and Zabih [76] relaxed these conditions to the above developed.

Optimality and Termination

Let us briefly discuss termination and optimality of the two move-making algorithms. Regarding termination, Veksler [111] proved that both terminate after a number of at most $O(n)$ iterations,

[12] Recall that the binary Boolean function Ψ_{ij} is submodular if and only if $\Psi_{ij}(1,1) + \Psi_{ij}(0,0) \leq \Psi_{ij}(0,1) + \Psi_{ij}(1,0)$ is fulfilled.

where n is the number of variables. However, Veksler states that in an experimental setup both algorithms converged after 2-8 iterations.

Regarding optimality, Boykov et al. [22], Veksler [111] give an upper bound on the MRF energy for the expansion algorithm if Ψ_{ij} is a metric. To this end, suppose that Ψ_{ij} is a metric for all $(i,j) \in \mathcal{E}$ and let

$$c := \max_{(i,j)\in\mathcal{E}} \left(\frac{\max_{\alpha\neq\beta\in\mathcal{L}} \Psi_{ij}(\alpha,\beta)}{\max_{\alpha\neq\beta\in\mathcal{L}} \Psi_{ij}(\alpha,\beta)} \right) \tag{4.40}$$

denote the maximum ratio of the largest nonzero value to the smallest nonzero value of all priors.[13] Then, the following bound on the MRF energy can be proved:

Theorem 6 (Boykov et al. [22, Thm. 6.1])**.** *Let* $\hat{\mathbf{x}}$ *be a local minimum w.r.t. expansion moves and let* $\mathbf{x}^*$ *denote a global minimum. Then,* $E(\hat{\mathbf{x}}) \leq 2cE(\mathbf{x}^*)$*.*

For the ordinary Potts model it is easy to verify that $c = 1$ and therefore the expansion algorithm yields an approximation within a factor of two.

4.6 Minimizing Nonsubmodular Functions

In Section 4.5 we have introduced two major move-making algorithms, the expansion and the swap algorithm, and stated the conditions under which they are applicable. In this section we will concentrate on multilabel problems with nonsubmodular MRF energies as they often occur in practice. As already discussed, most interesting, e.g. discontinuity-preserving, multilabel MRF energies are NP-hard [22] and thus approximations are inevitable.

Recall that the previously seen move-making algorithms are based on the efficient computation of the transformation function via a minimum s-t cut. In this section, we present the so called *fusion move* introduced by Lempitsky et al. [81, 82, 83], which is the natural generalization of both the expansion and the swap move. The key idea is to allow any two arbitrary labelings $\mathbf{x}^0$ and $\mathbf{x}^1$ to be "fused" and to efficiently compute the solution by a graph cut. However, the standard graph cut technique from Section 4.3 can not be applied if the fusion energy is nonsubmodular. Rother et al. [96] for instance proposed the truncation of nonsubmodular terms such that each term satisfies submodularity.

In this section, we present the fusion move and the above mentioned BHS algorithm, which will allow us to deal with nonsubmodular fusion energies. Before we continue, let show that even nonsubmodular Boolean MRF energies are NP-complete in general.

NP-completeness of Nonsubmodular Energies

In the following we show that even for a binary set of labels, i.e. $\mathcal{L} := \mathbb{B}$, and nonsubmodular priors, the problem is NP-complete. Therefore, let us consider the following decision problem PAIRWISE ENERGY-MIN:

[13] Note that the c is well-defined, since Ψ_{ij} is a metric and thus for $\Psi_{ij}(\alpha,\beta) \neq 0$.

Instance:	A nonsubmodular pseudo-Boolean function $E(\mathbf{x}) : \mathbb{B}^n \to \mathbb{R}$ with terms depending on up to two variables, and a constant K.
Question:	Does there exist a configuration $\mathbf{x} \in \mathbb{B}^n$ such that $E(\mathbf{x}) \leq K$?

For the proof we will use a well-known concept from graph theory:

Definition 15. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a subset $S \subseteq \mathcal{V}$ is a **vertex cover** if for every edge $(u, v) \in \mathcal{E}$ at least one of u and v is contained in S.*

Consider the corresponding decision problem VERTEX COVER, which is defined as follows:

Instance:	An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a constant K.
Question:	Does $\mathcal{G}$ have a vertex cover S of size at most K?

It is well known that VERTEX COVER is NP-complete [46]. Now, let us show the following claim.

Claim 3. PAIRWISE ENERGY-MIN *is* NP*-complete.*

Proof. Membership in NP follows immediately from a guess-and-check procedure, since a certificate, i.e. a configuration $\mathbf{x} \in \mathbb{B}^n$, can be checked in polynomial time as follows: given $\mathbf{x}$, evaluating each term in $E(\mathbf{x})$ and computing the sum requires time $O(|E|)$, where $|E|$ is the number of terms.[14] For the hardness, we give a polynomial-time many-one reduction VERTEX COVER $\leq_p$ PAIRWISE ENERGY-MIN and show the correctness.

Let $\mathcal{I} := (\mathcal{G}, K)$ be an instance of VERTEX COVER. We define an instance $\mathcal{J} := (E, K)$ of PAIRWISE ENERGY-MIN as follows: for each $i \in \mathcal{V}$ create a variable x_i and define a corresponding function Φ_i as

$$\Phi_i(x_i) := x_i,$$

and for every edge $(i, j) \in \mathcal{E}$ we define

$$\Psi_{ij}(x_i, x_j) := \begin{cases} K + 1 & \text{if } x_i = x_j = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Now, we define the energy function E as

$$E(\mathbf{x}) := \sum_{i \in \mathcal{V}} \Phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i, x_j).$$

Clearly, the reduction can be done in polynomial time in the size of the graph $\mathcal{G}$, since the number of terms in E is bound by $O(|V| + |E|)$ and the constructed PBF is nonsubmodular (cf. Section 4.2), since

$$\Psi_{ij}(0, 0) + \Psi_{ij}(1, 1) - \Psi_{ij}(0, 1) - \Psi_{ij}(1, 0) > 0. \tag{4.41}$$

[14] We make the additional assumption that each term can be evaluated in time polynomial in the size of the input, e.g. by looking up the value in a table which is part of the input or by computing some (algebraic) function in P.

Finally, let us show that

$$\mathcal{I} \in \text{VERTEX COVER} \Leftrightarrow \mathcal{J} \in \text{PAIRWISE ENERGY-MIN}.$$

($\Rightarrow$): Suppose that $\mathcal{I} \in$ VERTEX COVER. Then, $S \subseteq \mathcal{V}$ denotes a vertex cover of size at most K. Now define for every $i \in \mathcal{V}$

$$x_i := \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{else.} \end{cases}$$

Clearly, every x_i is well-defined and $E(\mathbf{x}) \leq K$.

($\Leftarrow$): Suppose that $\mathcal{J} \in$ PAIRWISE ENERGY-MIN. Since $E(\mathbf{x}) \leq K$, no term $\Psi_{ij}(x_i, x_j)$ in E exists such that both variables are assigned zero. Thus, we define for every $i \in \mathcal{V}$,

$$i \in S \Leftrightarrow x_i = 1,$$

and observe that S is a subset of $\mathcal{V}$ and a vertex cover of size at most K and the claim follows.

□

It is noteworthy that Kolmogorov and Zabih [76] established a similar reduction from the INDEPENDENT SET problem.

Fusion Move

In the beginning of this section we have informally already discussed the idea of “fusing” two labelings. The basic concept is to combine two suboptimal labelings, which we will refer to as *proposals*, such that the result in general may contain optimal parts of both proposals and is of lower or at least is of equal energy.

Given two configurations $\mathbf{x}^0, \mathbf{x}^1 \in \mathcal{L}^n$ and a Boolean vector $\mathbf{y} \in \mathbb{B}^n$, a vector $\mathbf{x}^c \in \mathcal{L}^n$ is said to be a *fusion* of $\mathbf{x}^0$ and $\mathbf{x}^1$ if

$$\mathbf{x}^c(\mathbf{y}) := \mathbf{x}^0 \circ (1 - \mathbf{y}) + \mathbf{x}^1 \circ \mathbf{y}, \tag{4.42}$$

where $\circ$ again denotes the Hadamard product (cf. Section 4.5). Often this combination is denoted as $\mathbf{x}^c = \mathbf{x}^0 \odot \mathbf{x}^1$ and is referred to as a *fusion move*. The great importance of the approach lies in the ability to combine *any* two labelings. It is easy to see that the expansion move and the swap move are special cases of the fusion move.

The resulting Boolean energy has already been stated earlier (cf. Section 4.5, Eq. (4.37)):

$$E(\mathbf{x}^c(\mathbf{y})) = \sum_{i \in \mathcal{V}} \Phi_i(x_i^c(y_i)) + \sum_{(i,j) \in \mathcal{E}} \Psi_{ij}(x_i^c(y_i), x_j^c(y_j)). \tag{4.43}$$

In case the resulting fusion energy is submodular, the global minimum can be computed with a minimum cut as seen in Section 4.3. In the next section we will see how to deal with nonsubmodular energies.

In addition, the fusion operation it is not necessarily limited to a discrete (one-dimensional) space of labels. In fact, $\mathcal{L}$ may be any discrete or a continuous (e.g. $\mathcal{L} \subseteq \mathbb{R}^d$) space of labels. The latter is for instance is used to determine the optical flow between two images [82].

```
Input: initial labeling $\mathbf{x}^i$.
Result: local minimum $\mathbf{x}^c$.
1  $\mathbf{x}^0 \leftarrow \mathbf{x}^i$;
2  repeat
3      $\mathbf{x}^1 \leftarrow \text{generateProposal}(\mathbf{x}^i, \mathbf{x}^0)$;
4      $\tilde{\mathbf{x}} \leftarrow \arg\min E(\mathbf{x}^0 \odot \mathbf{x}^1)$;
5      if $\tilde{\mathbf{x}}$ is partial then
6          $\mathbf{x}^c \leftarrow \tilde{\mathbf{x}} \lhd \arg\min_{\mathbf{c} \in \{\mathbf{x}^0, \mathbf{x}^1\}} E(\mathbf{c})$;
7      else
8          $\mathbf{x}^c \leftarrow \tilde{\mathbf{x}}$;
9      end
10 until $maxIterations$ reached;
```

Algorithm 4.3: Pseudocode for the fusion move algorithm.

The generic form of the fusion algorithm is outlined in Algorithm 4.3. As with the expansion and the swap algorithm, the fusion algorithm is very simple and works as follows: starting from an initial configuration $\mathbf{x}^0$, in each iteration a proposed labeling $\mathbf{x}^1$ is generated (line 3). We will later explain how we generate such proposals. Then, the optimal fusion move, denoted by $\mathbf{x}^0 \odot \mathbf{x}^1$, w.r.t. the Boolean fusion energy is computed with the BHS algorithm (line 4), which computes a partial labeling (in the next section we will see the details). In case that the fusion energy is nonsubmodular, the BHS algorithm returns only a partial labeling $\tilde{\mathbf{x}}$, which is then merged (the operation is denoted by $\lhd$) with the proposal which has lower energy. Note that the procedure always accepts solutions.

Clearly, the efficiency of this steepest descent algorithm depends on the capability of generating valuable proposals. Lempitsky et al. [83] state that in an experimental setting they observed that the number of unlabeled variables correlates with the number of nonsubmodular terms in the fusion energy. Informally, a submodular fusion term favors neighboring pixels from the same proposal (cf. Example 5). Thus, we conclude that proposals should be smooth w.r.t. the priors and result in submodular fusion energies.

BHS Algorithm

In the previous section we have stated that the global optimum of a fusion move can be computed if the Boolean energy (4.43) is submodular. However, if not, a concept from *Quadratic Pseudo-Boolean Optimization* (QPBO), the so called *BHS Algorithm*, allows the computation of a partially optimal solution $\mathbf{x} \in \mathbb{B}^n$. The method was developed by Hammer et al. [52] and, informally, such a partial solution can be computed by obtaining the strongest possible lower bound, the *master roof*, of a quadratic pseudo-Boolean function. Here, partiality of a solution means that each x_i takes values from the set $\{0, 1, ?\}$, where "?" can be considered as "unknown" or "unlabeled". Boros et al. [12][15] were the first to state a network flow-based implementation of the method and, in reference to the authors, the method is referred to as the BHS algorithm.

[15]Unfortunately, it was impossible to us and others in that field to obtain the original publication. We refer the reader to Boros and Hammer [11], which apparently contains most of the work but omits the proofs.

Rother et al. [99, 99] and Kolmogorov and Rother [72], who introduced this method to the field of computer vision, state the properties of the algorithm as follows:

1. Persistency: given a complete labeling $\mathbf{y}$ and a fusion $\mathbf{z}$ of $\mathbf{x}$ and $\mathbf{y}$ defined as: $z_i := x_i$ if $x_i \in \{0, 1\}$ and $z_i := y_i$ else. Then, $E(\mathbf{z}) \leq E(\mathbf{y})$.

2. Partial optimality: for at least one global minimum $\hat{\mathbf{x}}$ of the fusion energy (4.43), $x_i = \hat{x}_i$ if x_i is labeled, i.e. $x_i \in \{0, 1\}$.

3. In case all terms of energy (4.43) are submodular, all variables will be labeled.

4. The algorithm is invariant to "flippings": for a subset of variables the meaning of 0 and 1 can be swapped (transforming nonsubmodular terms into submodular and vice versa).

The strength of the method clearly depends on the number of variables that are labeled. However, since we have seen that the problem of minimizing nonsubmodular energies is NP-hard in general, not all variables will be labeled if the energy contains nonsubmodular terms [72]. Moreover, Rother et al. [98, 99] developed several heuristics to deal with the unlabeled variables.

4.7 Higher-order Energy Functions

So far, we have only discussed first-order MRF energies with terms of arity at most two. However, such energies often do not capture the rich characteristics of natural scenes and for some applications it is preferable to consider higher-order clique potentials where the cliques are of size three or more. Often such energies are referred to as *higher-order* energy functions.

Kolmogorov and Zabih [76] stated a reduction from a submodular second-order Boolean energy to a first-order Boolean energy which preserves submodularity. Later, Freedman and Drineas [42] restated the reduction as an algebraic formula. To the best of our knowledge, all reductions introduce new variables to the energy. Over the years, many reductions have been proposed and we refer the reader to Ishikawa [57] for a detailed analysis.

4.8 Maximum-Flow Algorithms for Computer Vision

Even though the standard algorithms for computing maximum flows, and equivalently minimum cuts, in the constructed networks operate in strongly polynomial time, algorithms which take advantage of the underlying structure of the graph have shown to outperform known strongly polynomial-time algorithms in an experimental setting [16, 18, 48]. Several new algorithms and adaptions have been proposed (see for instance Arora et al. [3], Delong and Boykov [36], and Liers and Pardella [85]).

Moreover, in a setting where the structure of the constructed flow network does not change much (e.g. in interactive image segmentation), the network flow can be computed incrementally (see for instance Boykov and Jolly [15]). For further details on dynamic graph cut algorithms we refer to the work of Kohli [68], Kohli and Torr [69, 70].

4.9 Image Denoising with Graph Cuts

In Section 2.1, we have seen a mathematical formulation of continuous and discrete images, and of a major form of degradation (noise) which may appear in the process of image acquisition. In the course of this work, we focus on random degradation which can be described with the presented noise models (cf. Section 2.1). The process of removing the degradation is referred to as *image restoration* and in particular we speak of image *denoising* whenever we want to express that the degradation is caused by noise. The quality of the restored result depends on the chosen noise and parameter model which should be as close as possible to reality. In this chapter, we discuss *regularization* as a variational method for *inverse problems* such as image denoising.

Let $u_0 : \Omega \to \mathbb{R}$ be the intensity function of an original image and let $u^\delta : \Omega \to \mathbb{R}$ be the intensity function of the observed (degraded) image. Given a noisy image u^δ, the problem of finding u_0 from u^δ is referred to as *image denoising*. Reconstructing u_0 from u^δ is a difficult task since often little is known about the noise. For instance, it might be the case that only statistical measures such as the mean or the variance of the noise can be inferred.

A common approach for finding a solution to the above problem is the so called *Tikhonov regularization* [110], which is to minimize the functional

$$\mathcal{T}_{\alpha,u^\delta}(u) := \mathcal{S}(u, v^\delta) + \alpha\mathcal{R}(u), \tag{4.44}$$

where $\alpha > 0$ is the *regularization parameter*, and $\mathcal{R}$ is a non-negative functional. The first term is called the *fit-to-data term* and measures the fitting of the (restored) solution u to the observed data u^δ whereas the second term is referred to as the *regularization term* and accounts for the "smoothness" or variation of the solution u. The regularization parameter α controls the tradeoff between the two terms. If α is chosen small, the solution tends to fit closely to the observed data u^δ and for large α the solution tends to be smooth.

A common choice for the data fitting term $\mathcal{S}$ is the L^1 or the (squared) L^2 metric whereas the regularization term $\mathcal{R}$ is chosen to depend on the gradient ∇u of the solution. For a better understanding, let us consider a concrete example taken from Aubert and Kornprobst [4, Eq. 3.4]:

Example 9. *Tikhonov and Arsenin [110] proposed minimizing the functional*

$$\mathcal{T}_{\alpha,u^\delta}(u) := \int_\Omega |u - u^\delta|^2 d\mathbf{x} + \alpha \int_\Omega |\nabla u|^2 d\mathbf{x}, \tag{4.45}$$

where ∇u denotes the gradient of u (cf. Section 2.1). Depending on α, a minimum of the above functional prefers solutions with a low gradient such that the noise is removed.

For denoising applications, it is common to consider minimizing (convex) first-order functionals of the general form

$$\mathcal{S}(u, u^\delta) + \alpha\mathcal{R}(u) := \int_\Omega \phi(u, u^\delta) d\mathbf{x} + \alpha \int_\Omega \psi(\mathbf{x}, u, \nabla u) d\mathbf{x}, \tag{4.46}$$

where ϕ is some *fit-to-data function* and $\alpha > 0$ [103]. We speak of *first-order regularization* if ∇u is the highest order differentiation of u which ψ depends on. In the course of this work, we consider this type only. In general, a first-order regularization model is said to be *isotropic* if

$$\psi(\mathbf{x}, u, \nabla u) = \widehat{\psi}(\mathbf{x}, u, |\nabla u|), \tag{4.47}$$

i.e. the function ψ does not depend on the orientation of the gradient. Otherwise, a model is said to be *anisotropic*.

Let us discuss some choices for ϕ and the resulting discrete forms of $\mathcal{S}$. In case the fit-to-data function is defined as the absolute difference, i.e. $\phi(u, u^\delta) := |u - u^\delta|$, we obtain the discrete L^1 metric

$$\int_\Omega |u - u^\delta| \approx \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |u_{ij} - u_{ij}^\delta| \tag{4.48}$$

for discrete images $\mathbf{u}$ and $\mathbf{u}^\delta$. If ϕ is defined as the squared differences, i.e. $\phi(u, u^\delta) := |u - u^\delta|^2$, we get the discrete form

$$\int_\Omega |u - u^\delta|^2 \approx \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} |u_{ij} - u_{ij}^\delta|^2, \tag{4.49}$$

which is the squared L^2 metric.

In the following, we will discuss various forms of ψ and their discrete counterparts used for first-order regularization. For each functional we investigate the applicability of the graph cut methods we have discussed. In Chapter 5 we present experimental results for some selected functionals.

4.10 First-Order Regularization Functionals for Denoising

In this section, we discuss (convex) first-order regularization methods for denoising summarized by Scherzer et al. [103, Sec. 4] and the corresponding finite-dimensional discrete forms. For each discrete energy, we investigate the applicability of the graph cut methods. First, let us recall the conditions for the applicability. For the expansion algorithm we required in Section 4.5 for all priors Ψ_{ij}

$$\Psi_{ij}(\alpha, \alpha) + \Psi_{ij}(\beta, \gamma) \leq \Psi_{ij}(\beta, \alpha) + \Psi_{ij}(\alpha, \gamma), \quad \forall \alpha, \beta, \gamma \in \mathcal{L}, \tag{4.50}$$

whereas the condition for the swap algorithm is

$$\Psi_{ij}(\alpha, \alpha) + \Psi_{ij}(\beta, \beta) \leq \Psi_{ij}(\alpha, \beta) + \Psi_{ij}(\beta, \alpha), \quad \forall \alpha, \beta \in \mathcal{L}, \tag{4.51}$$

and submodularity for an n-ary prior Ψ_C is defined as

$$\Psi_C(\mathbf{x} \vee \mathbf{y}) + \Psi_C(\mathbf{x} \wedge \mathbf{y}) \leq \Psi_C(\mathbf{x}) + \Psi_C(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathcal{L}^n, \tag{4.52}$$

where $\mathbf{x} \vee \mathbf{y}$ ($\mathbf{x} \wedge \mathbf{y}$) denotes the componentwise maximum (minimum) of $\mathbf{x}$ and $\mathbf{y}$. These conditions must hold for all i, j and C, respectively.

Total Variation

In their work Rudin, Osher, and Fatemi [101] (ROF) introduced *total variation* (TV)-based denoising models. The total variation of a continuous image u is stated as

$$\|u\|_{TV} := \int_\Omega |\nabla u| d\mathbf{x} \tag{4.53}$$

and has widely been used as an edge preserving regularizer Chan et al. [24]. The ROF model for regularization is then written as

$$\int_\Omega |u - u^\delta|^2 d\mathbf{x} + \alpha \int_\Omega |\nabla u| d\mathbf{x}. \tag{4.54}$$

The first term is the squared L^2 metric of the difference between the solution u and the observed image u^δ and requires u to preserve a significant part of the features from the observed image. The second term is the total variation of u and accounts for the amount of oscillation in the image such that noise effects diminish. It is noteworthy that the choice of the data term assumes additive Gaussian noise with zero mean as a noise model.

A common approach is to first state the above functional as a discrete finite-dimensional energy and then to optimize the energy. According to Chan et al. [24], common discrete forms of (4.53) are

$$\|\mathbf{u}\|_{TV} = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y-1} \left(|u_{i+1,j} - u_{ij}| + |u_{j,j+1} - u_{ij}|\right), \text{ and} \tag{4.55}$$

$$\|\mathbf{u}\|_{TV} = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y-1} \sqrt{(u_{i+1,j} - u_{ij})^2 + (u_{j,j+1} - u_{ij})^2}, \tag{4.56}$$

where again $\mathbf{u}$ is a discrete image.

The form (4.55) is a discrete form of the *anisotropic* TV $\int_\Omega (|u_x| + |u_y|) d\mathbf{x}$ and (4.56) is a discretization of the *isotropic* TV $\int_\Omega \sqrt{u_x^2 + u_y^2} d\mathbf{x}$, where here the functions u_x and u_y denote the partial derivatives of u with regards to x and y, respectively. It is noteworthy that the isotropic TV is rotational invariant whereas the anisotropic TV is not. Informally, the anisotropic TV prefers edges or boundaries along the directions of the image axis.

Let us investigate the applicability of graph cut algorithms. Given an image graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ arising from a regular grid, equation (4.55) can be brought into the form

$$\sum_{(i,j) \in \mathcal{E}} \Psi(x_i, x_j), \tag{4.57}$$

with $\Psi(x_i, x_j) := |x_i - x_j|$. Since Ψ is convex in $\delta := x_i - x_j$ it can be minimized exactly with the construction suggested by Ishikawa [56] (cf. Section 4.4) and with the construction by Darbon [30] (cf. Section 4.4), and approximated with both the expansion and the swap algorithm.

For further details on discrete forms of the anisotropic total variation and formulations through level sets we refer the reader to the work of Chambolle and Darbon [23], Darbon [29, 30], Darbon and Peyronnet [31], Darbon and Sigelle [32, 33, 34, 35].

However, equation (4.56) can not directly be stated as a sum of pairwise potentials (except for the case that $\mathcal{L} := \mathbb{B}$ where we can easily construct an equivalent PBF). The energy needs to be formulated as a clique potential with three parameters, i.e.

$$\sum_{(a,b,c)\in\mathcal{C}'_3} \Psi(x_a, x_b, x_c), \tag{4.58}$$

where $\mathcal{C}'_3$ denotes the set of cliques of size three such that we have three corresponding pixels $(u_{ij}, u_{i+1,j}, u_{i,j+1})$ which, basically, form a triangle. In the following we will show that for $\mathcal{L} := \mathbb{B}$ the submodularity condition (4.52) is satisfied. Unfortunately, a label space where $\mathcal{L}$ are consecutive integers with $|\mathcal{L}| > 2$ does not fulfill submodularity and thus, to the best of our knowledge, no reduction to a quadratic submodular function exists. Let us show that these claims hold.

Proposition 3. *For $\mathcal{L} := \mathbb{B}$, the clique potential*

$$\Psi(x_a, x_b, x_c) := \sqrt{(x_a - x_b)^2 + (x_a - x_c)^2} \tag{4.59}$$

is submodular.

Proof. Let $\mathbf{x}, \mathbf{y} \in \mathbb{B}^3$. It is easy to see that if for the two vectors $\mathbf{x} \leq \mathbf{y}$ or $\mathbf{x} \geq \mathbf{y}$, or $\mathbf{x} = \overline{\mathbf{y}}$ holds, then submodularity is fulfilled, i.e.

$$\Psi(\mathbf{x} \vee \mathbf{y}) + \Psi(\mathbf{x} \wedge \mathbf{y}) = \Psi(\mathbf{x}) + \Psi(\mathbf{y}) \tag{4.60}$$

holds. Thus, we only have to consider vectors $\mathbf{x}, \mathbf{y} \in \mathbb{B}^3$ where indices $i \neq j$ exist such that $x_i < y_i$ and $x_j > y_j$ (or the converse) and $\mathbf{x} \neq \overline{\mathbf{y}}$. Informally, these are all pairs of noncomplementary vectors such that the elementwise minimum and maximum contain elements of both vectors. For a better understanding let us write the vectors as $\begin{pmatrix} x_a & x_b & x_c \\ y_a & y_b & y_c \end{pmatrix}$. The remaining cases we have to consider are of the form:

$$\begin{pmatrix} v & u & \overline{u} \\ v & \overline{u} & u \end{pmatrix} \text{ and the symmetric forms } \begin{pmatrix} u & v & \overline{u} \\ \overline{u} & v & u \end{pmatrix} \text{ and } \begin{pmatrix} u & \overline{u} & v \\ \overline{u} & u & v \end{pmatrix}, \tag{4.61}$$

where $u, v \in \mathbb{B}$. Without loss of generality, let us assume that $u = 1$. For the first case we get the submodularity conditions

$$\sqrt{(v-u)^2 + (v-u)^2} + \sqrt{(v-\overline{u})^2 + (v-\overline{u})^2} \tag{4.62}$$
$$\leq \sqrt{(v-u)^2 + (v-\overline{u})^2} + \sqrt{(v-\overline{u})^2 + (v-u)^2}, \tag{4.63}$$

and for the second and the third case we get

$$\sqrt{(u-v)^2 + (u-u)^2} + \sqrt{(\overline{u}-v)^2 + (\overline{u}-\overline{u})^2} \tag{4.64}$$
$$\leq \sqrt{(u-v)^2 + (u-\overline{u})^2} + \sqrt{(\overline{u}-v)^2 + (\overline{u}-u)^2}, \tag{4.65}$$

and

$$\sqrt{(u-u)^2+(u-v)^2}+\sqrt{(\overline{u}-\overline{u})^2+(\overline{u}-v)^2} \tag{4.66}$$

$$\leq \sqrt{(u-\overline{u})^2+(u-v)^2}+\sqrt{(\overline{u}-u)^2+(\overline{u}-v)^2}. \tag{4.67}$$

Since $u-u=\overline{u}-\overline{u}=0$ and $u-\overline{u}=1$ the claim follows. $\square$

It is thus possible to exactly minimize these energies by first applying any of the reductions by Freedman and Drineas [42], Ishikawa [57], Kolmogorov and Zabih [76] to the quadratic case and then using the graph construction from Section 4.3. Next, let us consider the more interesting multilabel case.

Proposition 4. *For a label space of consecutive integers, i.e.* $\mathcal{L} := \{0,1,\ldots,k\}$*, the clique potential*

$$\Psi(x_a,x_b,x_c) := \sqrt{(x_a-x_b)^2+(x_a-x_c)^2} \tag{4.68}$$

is nonsubmodular for $k>1$.

Proof. By counterexample: let $k>1$, let $\mathbf{x} := (0,1,2)^T$, and let $\mathbf{y} := \mathbf{1}$ and observe that

$$\Psi(1,1,2)+\Psi(0,1,1) > \Psi(0,1,2)+\Psi(1,1,1) \tag{4.69}$$

violates the submodularity condition (4.52). $\square$

Finally, let us investigate the applicability of large moves to the multilabel case. Recall that for the expansion algorithm for any configuration $\mathbf{x}\in\mathcal{L}^3$ and for any label $\alpha\in\mathcal{L}$ the energy $\Psi(\mathbf{x}^c(\mathbf{y}))$ must be submodular. We use the equivalent submodularity definition given in Theorem 4.

Proposition 5. *For a label space of consecutive integers, i.e.* $\mathcal{L} := \{0,1,\ldots,k\}$*, the clique potential*

$$\Psi(\mathbf{x}^c(y_a,y_b,y_c)) := \sqrt{\left((x_a^c(y_a)-x_b^c(y_b)\right)^2+\left(x_a^c(y_a)-x_c^c(y_c)\right)^2} \tag{4.70}$$

is nonsubmodular for $k>1$.

Proof. By counterexample: let $k>1$, let $\alpha := 1$, and let $\mathbf{x} := (0,0,2)^T$. Then, by Theorem 4 the submodularity inequality must hold for every subset $X\subseteq V$ and distinct $i,j\in V\setminus X$, where the sets are induced by the expansion move. Let us choose as X the empty set, i as the element triggered by y_b and j as the element associated with y_c. Thus,

$$\Psi(x_a^c(0),x_b^c(0),x_c^c(0))+\Psi(x_a^c(0),x_b^c(1),x_c^c(1)) \tag{4.71}$$

$$\leq \Psi(x_a^c(0),x_b^c(0),x_c^c(1))+\Psi(x_a^c(0),x_b^c(1),x_c^c(0)) \tag{4.72}$$

must hold. Keeping in mind that the expansions induced by $\mathbf{x}^c$ are

$$\Psi(x_a,x_b,x_c)+\Psi(x_a,\alpha,\alpha) \leq \Psi(x_a,x_b,\alpha)+\Psi(x_a,\alpha,x_c) \tag{4.73}$$

and by our choice of α and $\mathbf{x}$ we get

$$\Psi(0,0,2)+\Psi(0,1,1) \leq \Psi(0,0,1)+\Psi(0,1,2), \tag{4.74}$$

which clearly does not hold. Therefore, $\Psi(\mathbf{x}^c(\mathbf{y}))$ is nonsubmodular for $k>1$. $\square$

Finally, we analyze whether we can still apply the swap algorithm.

Proposition 6. *For a label space of consecutive integers, i.e.* $\mathcal{L} := \{0, 1, \dots, k\}$*, the clique potential*

$$\Psi(\mathbf{x}^c(\mathbf{y})) := \sqrt{\left(x_a^c(y_a) - x_b^c(y_b)\right)^2 + \left(x_a^c(y_a) - x_c(y_c)\right)^2} \tag{4.75}$$

is submodular for $k > 1$.

Proof. Let $k > 1$, let $\alpha, \beta \in \mathcal{L}$ be arbitrary labels, and let $\mathbf{x} \in \mathcal{L}^3$. Then, again by Theorem 4, for every two components of $\mathbf{y}$ we need to check the swap condition. We first consider (y_b, y_c):

$$\Psi(x_a^c, \alpha, \alpha) + \Psi(x_a^c, \beta, \beta) \leq \Psi(x_a^c, \alpha, \beta) + \Psi(x_a^c, \beta, \alpha) \tag{4.76}$$

$$\sqrt{2(x_a^c - \alpha)^2} + \sqrt{2(x_a^c - \beta)^2} \leq 2\sqrt{(x_a^c - \alpha)^2 + (x_a^c - \beta)^2}. \tag{4.77}$$

Let $A := (x_a^c - \alpha)^2$ and let $B := (x_a^c - \beta)^2$. Then, since $A, B \geq 0$,

$$\sqrt{2A} + \sqrt{2B} \leq \sqrt{4(A + B)} \qquad \mid (\cdot)^2 \tag{4.78}$$

$$2A + 2\sqrt{2A}\sqrt{2B} + 2B \leq 4A + 4B. \tag{4.79}$$

It remains to show that $\sqrt{2A}\sqrt{2B} \leq A + B$:

$$\sqrt{2A}\sqrt{2B} \leq A + B \tag{4.80}$$

$$2\sqrt{A}\sqrt{B} \leq A + B \qquad \mid (\cdot)^2 \tag{4.81}$$

$$4AB \leq (A + B)^2 = A^2 + 2AB + B^2, \tag{4.82}$$

which holds true for $A, B \geq 0$. Next, let us consider (y_a, y_b). It is easy to see that

$$\Psi(\alpha, \alpha, x_c^c) + \Psi(\beta, \beta, x_c^c) \leq \Psi(\alpha, \beta, x_c^c) + \Psi(\beta, \alpha, x_c) \tag{4.83}$$

$$\sqrt{(\alpha - x_c^c)^2} + \sqrt{(\beta - x_c^c)^2} \leq \sqrt{(\alpha - \beta)^2 + (\alpha - x_c^c)^2} + \sqrt{(\beta - \alpha)^2 + (\beta - x_c^c)^2} \tag{4.84}$$

holds. For symmetry reasons, (y_a, y_c) can be shown analogously. $\square$

Thus, we have shown that for discrete energies of the form (4.56) the swap algorithm can be applied and therefore the cubic transformation energy can efficiently be reduced to a submodular quadratic form by any reduction from Section 4.7.

Isotropic Regularization

A common isotropic first-order-regularization model is [94, 105]

$$\widehat{\psi}(\mathbf{x}, u, |\nabla u|) := |\nabla u|^2. \tag{4.85}$$

The discretization again is straightforward and we obtain

$$\|\mathbf{u}\|^2 = \sum_{i=1}^{n_x - 1} \sum_{j=1}^{n_y - 1} (u_{i+1,j} - u_{ij})^2 + (u_{i,j+1} - u_{ij})^2, \tag{4.86}$$

which on an image graph $\mathcal{G}$ can be stated as a pairwise potential (4.57) with $\Psi(x_i, x_j) := (x_i - x_j)^2$. It is not hard to see that for the Boolean case the same properties as for the anisotropic TV hold. Let us classify the function and see what algorithms we might be able to apply for the multilabel case.

Proposition 7. *For a label space of consecutive integers $\mathcal{L} := \{0, 1, \ldots, k\}$ and $k > 1$ the potential $\Psi(x_i, x_j) := (x_i - x_j)^2$ cannot be approximated with the expansion algorithm.*

Proof. By counterexample: let $k > 1$ and let $\alpha, \beta, \gamma \in \mathcal{L}$. Then,

$$\Psi(\alpha, \alpha) + \Psi(\beta, \gamma) \leq \Psi(\beta, \alpha) + \Psi(\alpha, \gamma) \tag{4.87}$$

$$0 + (\beta - \gamma)^2 \leq (\beta - \alpha)^2 + (\alpha - \gamma)^2 \tag{4.88}$$

$$0 \leq (\alpha - \beta)(\alpha - \gamma) \tag{4.89}$$

Now we choose $\gamma < \alpha < \beta$ or $\beta < \alpha < \gamma$ and observe that the inequality is violated. □

Finally, let us check the applicability of the swap algorithm.

Proposition 8. *For a label space of consecutive integers $\mathcal{L} := \{0, 1, \ldots, k\}$ and $k > 0$ and the potential $\Psi(x_i, x_j) := (x_i - x_j)^2$ the swap algorithm can be applied.*

Proof. Let $k > 1$ and let $\alpha, \beta \in \mathcal{L}$ be arbitrary. Then,

$$\Psi(\alpha, \alpha) + \Psi(\beta, \beta) \leq \Psi(\alpha, \beta) + \Psi(\beta, \alpha) \tag{4.90}$$

$$0 \leq 2(\alpha - \beta)^2 \tag{4.91}$$

holds and the claim follows. □

However, Ψ can be written as a convex function in $\delta := x_i - x_j$, thus the energy can be minimized exactly with both the constructions by Ishikawa [56] and Darbon [30].

Moreover, Scherzer et al. [103] list *weighted* regularization models, which aim at a better preserving of discontinuities in an image. As mentioned earlier, we want a model not to over-smooth edges. Such regions are characterized by a large gradient ∇u. Therefore, some weight depending on $\mathbf{x}$ is introduced which makes the regularization term small in areas where the gradient is large. The observed data can not directly be used for weight determination since noise might also cause regions with large gradients. Instead, a *mollifier* ρ is first applied to the recorded image u^δ with the goal of smoothing it. The measure $\nabla(\rho * u^\delta)$ can then be used to determine the weights. To this end, let $r \in \mathbb{R}$ and let $\lambda^2 > 0$. We define

$$g(r) := \frac{1}{1 + {}^{r^2}\!/_{\lambda^2}}. \tag{4.92}$$

The weighted isotropic regularization model for denoising is then

$$\widehat{\psi}(\mathbf{x}, u, |\nabla u|) := g(|\nabla(\rho * u^\delta)|)|\nabla u|^2. \tag{4.93}$$

It is readily seen that the model can directly be stated as the discrete energy

$$\sum_{i=1}^{n_x-1}\sum_{j=1}^{n_y-1} g_{ij}\left((u_{i+1,j}-u_{ij})^2+(u_{i,j+1}-u_{ij})^2\right), \tag{4.94}$$

where $\mathbf{g}$ is a matrix resulting from (4.93). Clearly, the energy can be defined on an image graph $\mathcal{G}$ and brought into the form $\Psi_{ij} := w_{ij}(x_i - x_j)^2$, with $\mathbf{w}$ being defined appropriately. The same properties we have established for the isotropic first-order regularization model (4.85) apply and the above energy can be minimized exactly with the constructions of Ishikawa [56] and Darbon [30]. Moreover, these energies can be approximated with the swap algorithm.

Analogous to the above model, the weighted ROF functional is defined as

$$\widehat{\psi}(\mathbf{x}, u, |\nabla u|) := \beta(\mathbf{x})|\nabla u|, \tag{4.95}$$

for some function $\beta : \Omega \to \mathbb{R}^+$. The finite-dimensional discrete form of functionals of the above form is similar to the discrete form of the isotropic TV with the only difference that the discrete energy must incorporate the weights and thus depend on $\mathbf{x} \in \Omega$. It is easy to see that with regard to graph cut applicability the same results as for the isotropic TV can be obtained.

Huber Model

Another prominent isotropic first-order regularization model is the so called *Huber* model [55, 78], which is defined as

$$\widehat{\psi}(\mathbf{x}, u, |\nabla u|) := \begin{cases} |\nabla u|^2 & \text{if } |\nabla u| \le 1, \\ 2|\nabla u| - 1 & \text{else.} \end{cases} \tag{4.96}$$

Clearly, the model is spatially dependent in the sense that in regions having a small gradient it penalizes less than in regions with a large gradient. Regarding discretization we rely on already established forms:

$$\Psi(x_a, x_b, x_c) := \begin{cases} (x_a - x_b)^2 + (x_a - x_c)^2 & \text{if } \sqrt{(x_a - x_b)^2 + (x_a - x_c)^2} \le 1, \\ 2\sqrt{(x_a - x_b)^2 + (x_a - x_c)^2} - 1 & \text{else.} \end{cases} \tag{4.97}$$

Let us for a moment assume our usual discrete space of labels $\mathcal{L} := \{0, 1, \ldots, k\}$ with $k > 0$, then in regions of the image where

$$|u_{ij} - u_{i+1,j}| + |u_{ij} - u_{i,j+1}| \le 1 \tag{4.98}$$

is satisfied the function Ψ takes the first and otherwise the second nature. Clearly, if (4.98) holds, then $\Psi \in \{0, 1\}$. Unless a discrete image satisfies (4.98) in all regions (e.g. a constant image), the established properties from the isotropic TV model inherit. Thus, with the presented methods the discrete Huber model (4.97) can be approximated with the swap algorithm only.

Anisotropic Regularization

So far, we have seen mostly isotropic regularization methods for denoising. Scherzer et al. [103] consider a class of *quadratic anisotropic first-order regularization* models which can be written as

$$\psi(\mathbf{x}, \nabla u) := \nabla u^T A(\mathbf{x}) \nabla u, \tag{4.99}$$

where A is a positive definite matrix defined as

$$A := \begin{pmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{pmatrix} \begin{pmatrix} g(|\nabla(\rho * u^\delta)|) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 & v_2 \\ -v_2 & v_1 \end{pmatrix}, \tag{4.100}$$

and $\mathbf{v} := (v_1, v_2)^T$ is defined as

$$\mathbf{v} := \begin{cases} \frac{\nabla(\rho * u^\delta)}{|\nabla(\rho * u^\delta)|} & \text{if } |\nabla(\rho * u^\delta)| > 0, \\ \mathbf{e}_1 & \text{else,} \end{cases} \tag{4.101}$$

where $\mathbf{e}_1$ is the canonical vector. It is readily seen that A is of the form

$$A = \begin{pmatrix} v_1^2 g(\cdot) + v_2^2 & v_1 v_2 g(\cdot) - v_1 v_2 \\ v_1 v_2 g(\cdot) - v_1 v_2 & v_2^2 g(\cdot) + v_1^2 \end{pmatrix}. \tag{4.102}$$

Figure 4.5 depicts the components of matrix A computed for the noisy image in Figure 4.4.

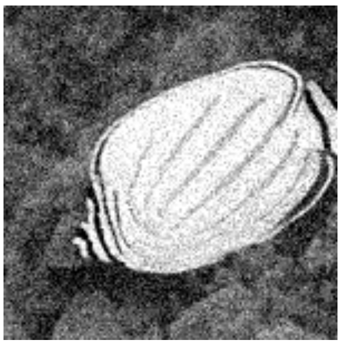


Figure 4.4: Noisy image.

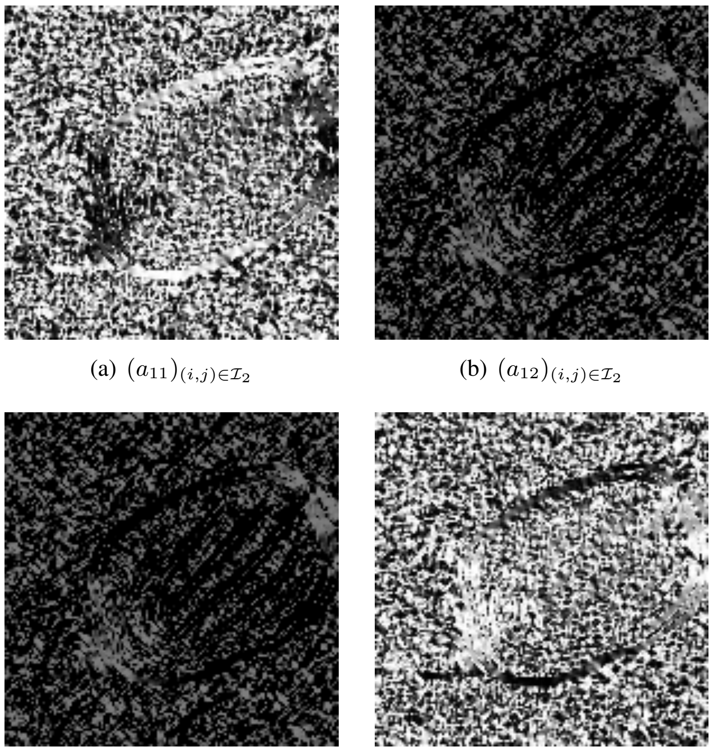
Let us investigate the properties of the above model.

$$\nabla u^T A(\mathbf{x}) \nabla u = \begin{pmatrix} u_x & u_y \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} \tag{4.103}$$

$$= a_{11} u_x^2 + (a_{12} + a_{21}) u_x u_y + a_{22} u_y^2, \tag{4.104}$$

where a_{ij} are constant and do not depend on u. Moreover, by (4.102) we have $a_{12} = a_{21}$. A discrete finite-dimensional form of (4.99) is

$$\sum_{i=1}^{n_x - 1} \sum_{j=1}^{n_y - 1} a_{11} u_x^2 + 2 a_{12} u_x u_y + a_{22} u_y^2. \tag{4.105}$$


(a) $(a_{11})_{(i,j)\in\mathcal{I}_2}$ (b) $(a_{12})_{(i,j)\in\mathcal{I}_2}$

(c) $(a_{21})_{(i,j)\in\mathcal{I}_2}$ (d) $(a_{22})_{(i,j)\in\mathcal{I}_2}$

Figure 4.5: Example for matrix A using mollifier $\boldsymbol{\rho} = {}^1\!/_9 \cdot J_3$.

Obviously, the first and the last term are independent pairwise potentials and thus the energy can be brought into the form

$$\sum_{(i,j)\in\mathcal{E}} \Psi_{ij}(x_i, x_j) + \sum_{(a,b,c)\in\mathcal{C}_3'} \Phi_{abc}(x_a, x_b, x_c), \tag{4.106}$$

where the pairwise potentials are defined as $\Psi_{ij}(x_i, x_j) := w_{ij}(x_i - x_j)^2$ with an appropriate weight matrix $\mathbf{w}$, and

$$\Phi_{abc}(x_a, x_b, x_c) := 2a_{12}(x_a - x_b)(x_a - x_c). \tag{4.107}$$

Earlier we stated that since Ψ_{ij} is submodular it can be minimized exactly. However, let us show the following.

Proposition 9. *For a discrete label space of consecutive integers, i.e. $\mathcal{L} := \{0, 1, \ldots, k\}$, the clique potential Φ_{abc} is nonsubmodular for $k \geq 1$ and $a_{12} = a_{21} \neq 0$.*

Proof. For $k = 1$ it is readily seen that

$$2a_{12}(x_a - x_b)(x_a - x_c) \tag{4.108}$$

is a pseudo-Boolean function of degree two and has both positive and negative coefficients. Thus, by Proposition 2, the energy (4.107) is nonsubmodular.

Even though the result immediately generalizes for $k > 1$ we give a justification. It is sufficient to show that no product cx_ix_j with $c > 0$ is submodular. To this end, let $\mathbf{x}, \mathbf{y} \in \mathcal{L}^2$. By the submodularity condition (4.52) we have

$$\max\{x_1, y_1\} \cdot \max\{x_2, y_2\} + \min\{x_1, y_1\} \cdot \min\{x_2, y_2\} \leq x_1x_2 + y_1y_2. \tag{4.109}$$

Let $\mathbf{x} := (1, 2)^T$ and let $\mathbf{y} := (2, 1)^T$ and observe that the above inequality is violated. $\square$

From our previous results, we can conclude immediately that the energy (4.106) can not be approximated with the expansion algorithm since not even the pairwise terms Ψ_{ij} can. Finally, let us show that the swap algorithm does not apply either.

Proposition 10. *For a discrete label space of consecutive integers, i.e.* $\mathcal{L} := \{0, 1, \dots, k\}$, *the clique potential* $\Phi_{abc}(\mathbf{x}^c(\mathbf{y})) := 2a_{12}(x_a^c(y_a) - x_b^c(y_b))(x_a^c(y_a) - x_c^c(y_c))$ *does not fulfill the swap condition for* $k \geq 1$ *and* $a_{12} = a_{21} \neq 0$.

Proof. By counterexample. Observe that

$$\Phi_{abc}(x_a^c, \alpha, \alpha) + \Phi_{abc}(x_a^c, \beta, \beta) \leq \Phi_{abc}(x_a^c, \beta, \alpha) + \Phi_{abc}(x_a^c, \alpha, \beta) \tag{4.110}$$

$$(x_a^c - \alpha)^2 + (x_a^c - \beta)^2 \leq 2(x_a^c - \alpha)(x_a^c - \beta). \tag{4.111}$$

Now, we choose $x_a^c = \alpha$ and $\beta \neq x_a^c$, and observe that the inequality is violated. $\square$

From the previous propositions we get as a consequence that the generalization which the quadratic anisotropic model introduces leaves us only with the fusion move for nonsubmodular energies.

Anisotropic Non-Quadratic Regularization

It is noteworthy that there exist a non-quadratic anisotropic first-order regularization model, which is stated as

$$\sqrt{\nabla u^T A(\mathbf{x}) \nabla u}. \tag{4.112}$$

Brought in a discrete form, we obtain

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sqrt{a_{11}u_x^2 + 2a_{12}u_xu_y + a_{22}u_y^2}. \tag{4.113}$$

In this particular case we are directly able to show that the swap condition is violated for some $\alpha, \beta \in \mathcal{L}$ and thus conclude that the function (4.113) is in general neither submodular nor does it fulfill the expansion condition.

Proposition 11. *For a discrete label space of consecutive integers, i.e. $\mathcal{L} := \{0, 1, \ldots, k\}$, the clique potential*

$$\Phi_{abc}(\mathbf{x}^c(\mathbf{y})) := \sqrt{a_{11}u_x^2 + 2a_{12}u_xu_y + a_{22}u_y^2} \tag{4.114}$$

with

$$u_x := x_a^c(y_a) - x_b^c(y_b), \tag{4.115}$$
$$u_y := x_a^c(y_a) - x_c^c(y_c). \tag{4.116}$$

does not fulfill the swap condition for $k \geq 1$.

Proof. Again, let us consider the condition where we fixed y_a, which is

$$\Phi_{abc}(x_a^c, \alpha, \alpha) + \Phi_{abc}(x_a^c, \beta, \beta) \leq \Phi_{abc}(x_a^c, \beta, \alpha) + \Phi_{abc}(x_a^c, \alpha, \beta), \tag{4.117}$$

and expands to

$$\sqrt{(a_{11} + 2a_{12} + a_{22})(x_a^c - \alpha)^2} + \sqrt{(a_{11} + 2a_{12} + a_{22})(x_a^c - \beta)^2} \tag{4.118}$$
$$\leq 2\sqrt{a_{11}(x_a^c - \alpha)^2 + 2a_{12}(x_a^c - \alpha)(x_a^c - \beta) + a_{22}(x_a^c - \beta)^2}. \tag{4.119}$$

It is easy to see that for a choice of $x_a^c = \alpha$, $x_a^c \neq \beta$, and coefficients

$$a_{11} + 2a_{12} + a_{22} > 4a_{22} \tag{4.120}$$

the inequality is violated and the claim follows. □

Summary

Let us summarize the results for the established discrete forms from the previous section in Table 4.1. For the exact solutions the constructions of both Ishikawa [56] and Darbon [30] can be applied.

#	Regularization	ψ	Discr. form	Exact	Exp.	Swap	Fusion
1	Anisotropic TV	$\|\nabla u\|$	(4.55)	✓	✓	✓	✓
2	Isotropic TV	$\|\nabla u\|$	(4.56)	✗	✗	✓	✓
3	Isotropic	$\|\nabla u\|^2$	(4.86)	✓	✗	✓	✓
4	Huber	see (4.96)	(4.97)	✗	✗	✓	✓
5	Weighted Isotropic	$g(\|\nabla(\rho * u^\delta)\|)\|\nabla u\|^2$	(4.94)	✓	✗	✓	✓
6	Weighted Iso. TV	$\beta(\mathbf{x})\|\nabla u\|$	cf. (4.56)	✗	✗	✓	✓
7	Anisotr.	$\nabla u^T A(\mathbf{x}) \nabla u$	(4.105)	✗	✗	✗	✓
8	Anisotr. non-quadr.	$\sqrt{\nabla u^T A(\mathbf{x}) \nabla u}$	(4.113)	✗	✗	✗	✓

Table 4.1: Summary of first-order regularization functionals w.r.t. graph cut applicability in the multilabel scenario.

CHAPTER 5

Experimental Results and Discussion

In Section 4.9 of the previous chapter, we have introduced various continuous first-order regularization models for the task of image denoising and stated discrete forms thereof. In this section, we discuss the experimental contribution of this thesis and present numerical results obtained by solving the discrete finite dimensional energies via graph cuts. On top of existing libraries we have built a Matlab implementation of the graph constructions which is able to deal with the discrete energies in both the Boolean and the multilabel MRF scenario. Moreover, it shall be stressed that from this point on we exclusively discuss discrete energies.

For our experiments with the expansion and the swap algorithm, we relied on the multilabel optimization framework developed by Boykov et al. [22].[1] Our implementation of the fusion move algorithm is built upon the BHS implementation by Rother et al. [98].[2] Furthermore, it is noteworthy that all frameworks make use of the algorithm developed by Boykov and Kolmogorov [18] to determine the maximum flow in the constructed network.[3] Finally, we emphasize that the constructed flow networks have integer capacities and thus may lead to small inaccuracies. All computations were run on an Intel Xeon E5520 (4 cores, 2.26GHz each) 12GB machine.

Even though we performed experiments for all of the discrete energies listed in Section 4.9, we only discuss and list the results for the following models:

1. anisotropic first-order regularization,
2. anisotropic total variational regularization, and
3. isotropic first-order regularization.

The first model (and its non-quadratic version) was approached with the fusion algorithm. The second and the third were minimized with the expansion and the swap algorithm, respectively.

[1]The source code of the framework is publicly available at `http://vision.csd.uwo.ca/code/`.
[2]Available at `http://pub.ist.ac.at/~vnk/software.html`.
[3]Also available at `http://pub.ist.ac.at/~vnk/software.html`.

In the first part of this chapter we explain our methodology with regard to test instance creation and result computation. In the second part, we present the conducted experiments and discuss the obtained results.

5.1 Methodology

Our experiments were performed on the basis of six commonly used test images from the field of computer vision. These images are: *Cameraman*, *Fish*, *House*, *Lenna*[4], *Mandrill*, and *Pirate*.[5] All images are grayscale images, i.e. $\mathcal{L} := \{0, 1, \ldots, 255\}$, and of size 150×150. The undistorted images and their histograms are depicted in Figures A.1 and A.2, respectively, in the Appendix.

For our experiments we created two sets of problem instances: the images in the first set were artificially degraded by *additive Gaussian noise* with parameters $\mu = 0$ and $\sigma^2 = 0.01$ whereas the test images in the second set were corrupted by *Salt & Pepper* noise with parameter $p = 0.1$. The full characteristics of the generated instances can again be found in Table A.1 and Table A.2 in the Appendix.

Let us denote by $\mathbf{u}^0$ the original (undistorted) discrete image, by $\mathbf{u}^\delta$ the noisy discrete image, and by $\mathbf{u}$ the restored discrete image, all of same size. For the input of each experiment we obtained the following characteristics:

1. L^1 norm $\|\mathbf{u}^0 - \mathbf{u}^\delta\|_1$,
2. L^2 norm $\|\mathbf{u}^0 - \mathbf{u}^\delta\|_2$,
3. avg. L^1 norm $\frac{\|\mathbf{u}^0 - \mathbf{u}^\delta\|_1}{\mathrm{len}(\mathbf{u}^0)}$,
4. avg. L^2 norm $\frac{\|\mathbf{u}^0 - \mathbf{u}^\delta\|_2}{\mathrm{len}(\mathbf{u}^0)}$,
5. relative error $r^\delta := \sum_p \frac{|u_p^0 - u_p^\delta|}{|\mathcal{L}|-1}$,
6. avg. relative error $\overline{r}^\delta := \frac{r}{\mathrm{len}(\mathbf{u}^0)}$,

where $\mathrm{len}(\mathbf{u}^0)$ is the number of elements in $\mathbf{u}^0$. From the results of each experiment we calculated:

1. L^1 norm $\|\mathbf{u}^0 - \mathbf{u}\|_1$,
2. L^2 norm $\|\mathbf{u}^0 - \mathbf{u}\|_2$,
3. avg. L^1 norm $\frac{\|\mathbf{u}^0 - \mathbf{u}\|_1}{\mathrm{len}(\mathbf{u}^0)}$,
4. avg. L^2 norm $\frac{\|\mathbf{u}^0 - \mathbf{u}\|_2}{\mathrm{len}(\mathbf{u}^0)}$,

[4]We refer the reader to http://www.cs.cmu.edu/~chuck/lennapg/ for the historical background of the image.

[5]Test images were taken from http://sipi.usc.edu/database/.

5. relative error $r := \sum_p \frac{|u_p^0 - u_p|}{|\mathcal{L}|-1}$,

6. avg. relative error $\overline{r} := \frac{r}{\mathrm{len}(\mathbf{u}^0)}$.

5.2 Anisotropic First-Order Regularization

In our first set of experiments, we choose to approximate the discrete forms of the anisotropic first-order regularization developed in Section 4.10. In order to minimize the discrete forms, we implemented the fusion move algorithm. Recall that the energies arising from anisotropic regularization in general result in nonsubmodular functions and thus can only be approximated.

In our first setting, we generated proposals from the initial image $\mathbf{u}^\delta$ by adding Gaussian white noise $\boldsymbol{\delta}$ and computing the *convolution* using the kernel matrix $\boldsymbol{\rho} := {}^1\!/_9 \cdot J_3$, where J_3 is a $3{\times}3$ matrix of ones. For details on the continuous and discrete convolution we refer to Scherzer et al. [103, Sec. 9.5] and Winkler [116, Sec. 1.2 and 2.3], respectively. The proposed image is then $\mathbf{x}^1 := \boldsymbol{\rho} * (\mathbf{u}^\delta + \boldsymbol{\delta})$ and should tend to generate smooth areas.

In a first run, we computed results for all combinations of data and regularization models for all six test images. The fit-to-data was either the L^1 or the squared L^2 norm of the difference between the restored and the observed image, i.e. $\mathbf{u} - \mathbf{u}^\delta$. The regularization term was chosen either to be the anisotropic (quadratic) or the anisotropic non-quadratic model. Due to the fast convergence of the fusion algorithm, the maximum number of iterations was set to 20.

Our initial parameters are listed in Table 5.1.

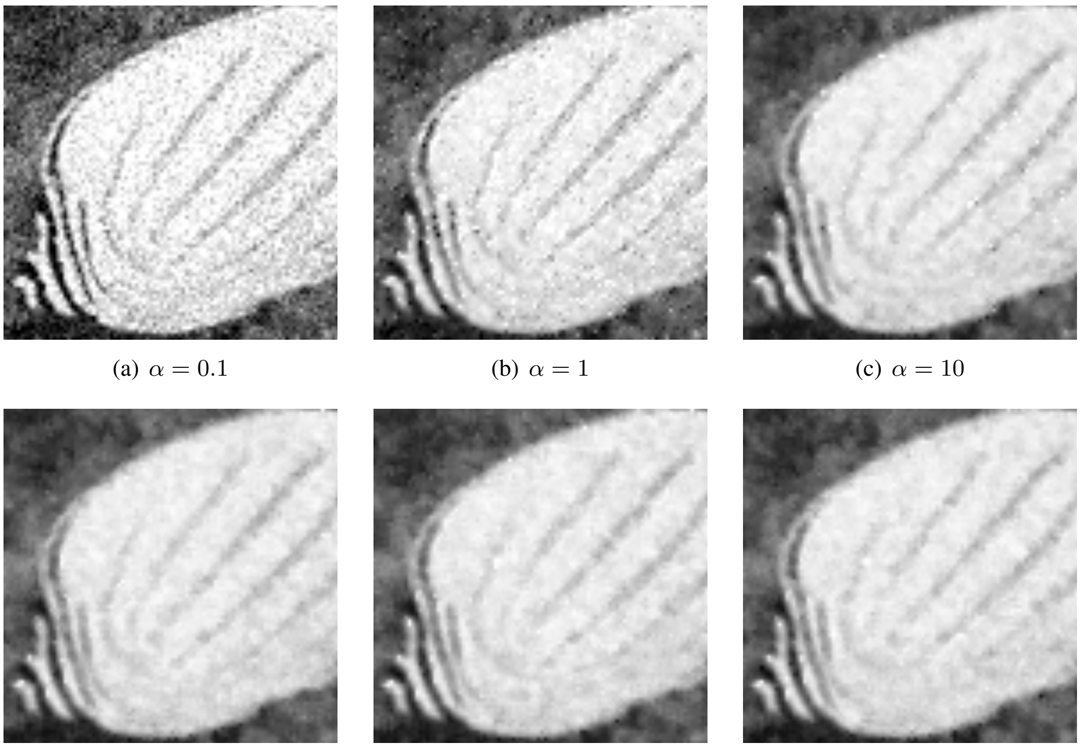
Parameter	Symbol	Value
Weight decay in $g(\cdot)$	λ^2	1
Convolution kernel	$\boldsymbol{\rho}$	${}^1\!/_9 \cdot J_3$
Mean of proposal noise $\boldsymbol{\delta}$	μ	0
Variance of proposal noise $\boldsymbol{\delta}$	σ^2	0.01

Table 5.1: Initial parameters.

In order to determine the optimal regularization parameter α for every image and every pair of data fidelity and regularization model, we obtained results for varying values of α. Starting from an initially small value, we chose α to increase by an order of magnitude in each iteration (the range of α was $\{10^{-4}, 10^{-3}, \dots, 10^{4}\}$). Figure 5.1 depicts restored images for increasing values of the regularization parameter. For our further experiments we used the value of α resulting in the lowest error regarding the L^2 distance between the restored image $\mathbf{u}$ and the uncorrupted original image $\mathbf{u}^0$. For the sake of brevity we list only the determined parameters for a subset of the test instances in Table 5.2. The full results are deferred to the Appendix.

However, we performed a series of experiments with the anisotropic regularization models on the second set of test instances, which were degraded with Salt & Pepper noise, and found that in our setting the model did not show capable of efficiently removing the noise.

In a second pass we ran our experiments again with the established parameters and adapted the mollifier $\boldsymbol{\rho}$ used for both the computation of the matrix A (in both anisotropic regularization


(a) $\alpha = 0.1$ (b) $\alpha = 1$ (c) $\alpha = 10$

(d) $\alpha = 100$ (e) $\alpha = 1000$ (f) $\alpha = 10000$

Figure 5.1: Varying values of α (squared L^2, anisotropic).

Data fidelity	Regularization	Cameraman	Fish	House	Lenna
L^1	Anisotropic	1	0.1	1	1
Squared L^2		10	10	10000	10
L^1	Anisotropic non-quadratic	10	10	100	10
Squared L^2		100	100	1000	1000

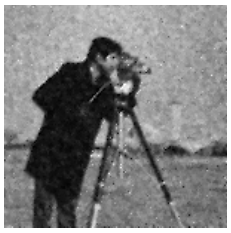
Table 5.2: Regularization parameter α determined for the tested models and instances.

models) and the generation of the proposals. For this run we used a Gaussian 3×3 kernel matrix. However, we found evidence that the Gaussian kernel matrix led to improvements both in the L^2 error of the restored image and visual quality. The best results were obtained from using the squared L^2 distance for the data term and a non-quadratic anisotropic regularization term. The series is depicted in Figure 5.3. It is noteworthy that in this case the number of maximum iterations was set to 100. On the other hand, we observed that Gaussian kernels of size larger than 10×10 led to blurred undesirable solutions.

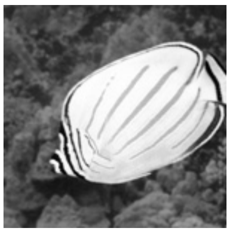
Moreover, we observed that in general the fusion algorithm tends to converge in our setting after approximately 20 iterations and when run longer does not significantly improve the result (cf. the typical curve in Figure A.3(a) in the Appendix). In addition, we point out that with our measurements we could not find any obvious criterion influencing the number of unlabeled variables returned by the BHS algorithm. Nevertheless, for negligible values of α the BHS algorithm returned a complete labeling as expected.

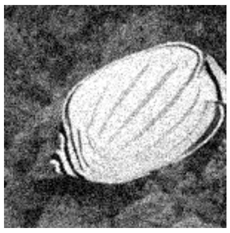
(a) Original $\mathbf{u}^0$

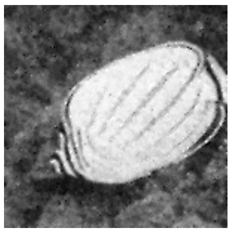
(b) Noisy image $\mathbf{u}^\delta$

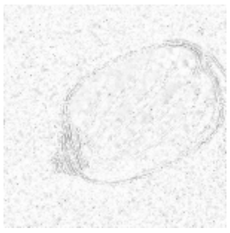

(c) Restored image $\mathbf{u}$

(d) Error $|\mathbf{u}^0 - \mathbf{u}|$


(e) Original $\mathbf{u}^0$

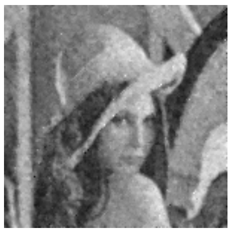

(f) Noisy image $\mathbf{u}^\delta$


(g) Restored image $\mathbf{u}$


(h) Error $|\mathbf{u}^0 - \mathbf{u}|$

(i) Original $\mathbf{u}^0$

(j) Noisy image $\mathbf{u}^\delta$


(k) Restored image $\mathbf{u}$

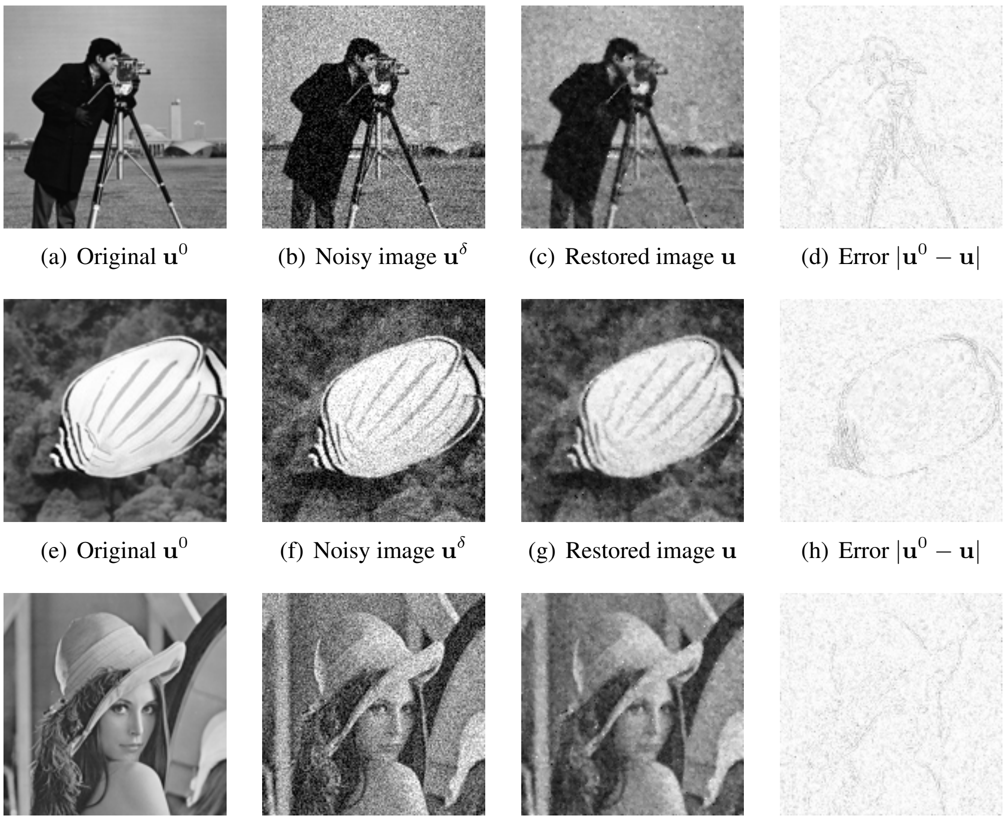
(l) Error $|\mathbf{u}^0 - \mathbf{u}|$

Figure 5.2: Gaussian degraded images restored by the anisotropic models.

5.3 Anisotropic Total Variational Regularization

In a second setting, we obtained results for both test sets by the discrete anisotropic total variation. Since the maximum flow algorithm our implementation is based on uses integer-valued flows we adapted the model such that for parameter values $\alpha < 1$ a multiplicative constant $\gamma := {}^1\!/\!_\alpha$ for the data term was introduced and α set to 1, preserving the ratio between the data and the regularization term.

This time we used for both test sets (Gaussian and Salt & Pepper noise) the L^1 distance for the data term. As verified in Section 4.10, both terms fulfill the expansion condition and thus can be approximated.


(a) Original $\mathbf{u}^0$ (b) Noisy image $\mathbf{u}^\delta$ (c) Restored image $\mathbf{u}$ (d) Error $|\mathbf{u}^0 - \mathbf{u}|$

(e) Original $\mathbf{u}^0$ (f) Noisy image $\mathbf{u}^\delta$ (g) Restored image $\mathbf{u}$ (h) Error $|\mathbf{u}^0 - \mathbf{u}|$

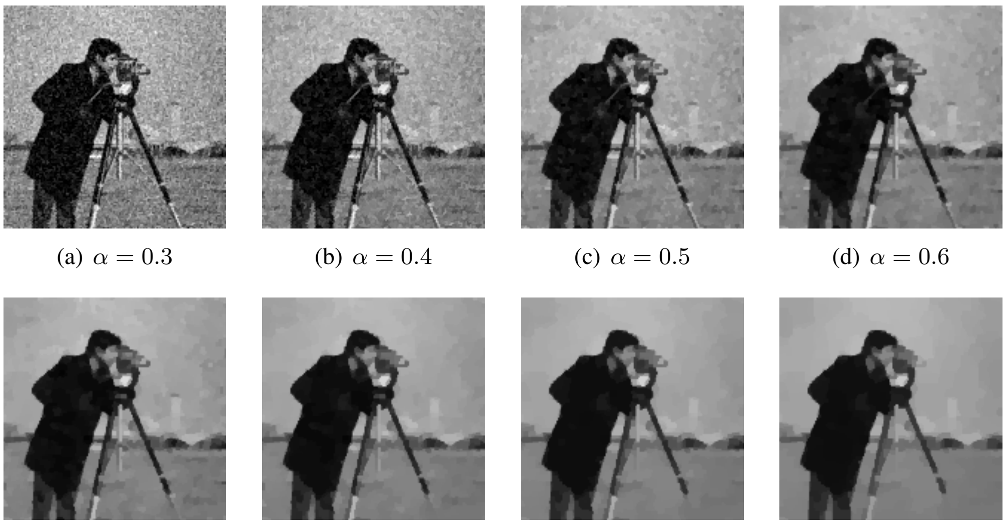
(i) Original $\mathbf{u}^0$ (j) Noisy image $\mathbf{u}^\delta$ (k) Restored image $\mathbf{u}$ (l) Error $|\mathbf{u}^0 - \mathbf{u}|$

Figure 5.3: Gaussian degraded images restored by the non-quadratic anisotropic model with a Gaussian kernel $\boldsymbol{\rho}$.

As before, we determined the regularization parameter α for every image resulting in the smallest L^2 error of the restored image. The tested range of α was $\{0.1, 0.2, \ldots, 1, 1.5, 2\}$. In Figure 5.4, we depict the inferred solutions for varying values of α. Figure A.3(b) in the Appendix illustrates the L^2 error of the solution versus increasing (log) values of α. The full results are again deferred to the Appendix.

Even though in our setting the anisotropic TV computed by the expansion algorithm is capable of efficiently smoothing large homogenous regions, it lacks in preserving small features (such as the camera handle, cf. Figure 5.4). It is known that truncating the priors may yield better results regarding discontinuities (cf. Veksler [111, 113]).

Figures 5.5 and 5.6 illustrate the results obtained with the anisotropic TV model. The rightmost image respectively depicts the difference between the restored and the original image. For better visibility we depict the complement image.


(a) $\alpha = 0.3$ (b) $\alpha = 0.4$ (c) $\alpha = 0.5$ (d) $\alpha = 0.6$

(e) $\alpha = 0.7$ (f) $\alpha = 0.8$ (g) $\alpha = 0.9$ (h) $\alpha = 1$

Figure 5.4: Varying values of α for anisotropic total variation and Gaussian noise.

5.4 Isotropic Regularization

In the third series of experiments, we obtained numerical results for the isotropic regularization model discussed in Section 4.9. As shown, the model can only be approximated with the swap algorithm. For our experiments we used the squared L^2 norm for the data term and iterated upon convergence (the maximum number of iterations was set to 100).

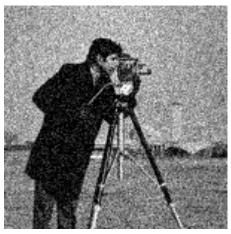
Again, we conducted experiments for both test sets and for several values of the regularization parameter. We observed that the model is not as sensitive as the anisotropic TV model regarding changes in the regularization parameter. Thus, we only include results for values of α from the set $\{0.1, 0.5, 1, 1.5, 2\}$.

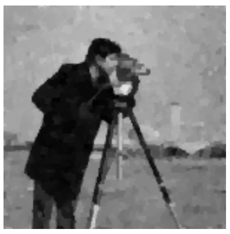
Figures 5.7 and 5.8 illustrate the obtained results. However, we found that the isotropic model tends to blur the solution and in particular is not capable of effectively removing Salt & Pepper noise.

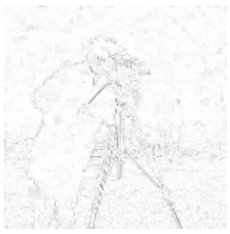
5.5 Extremely Degraded Images

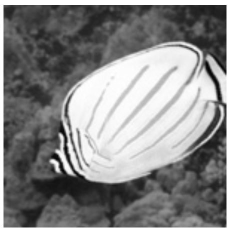
In order to demonstrate the power of the presented models, we chose to distort test images heavily with Salt & Pepper noise and then reconstructed the image with the anisotropic TV regularization model. Figure 5.9 shows the results. Even for an image which contained 90% noise the expansion algorithm is able to yield an acceptable solution.

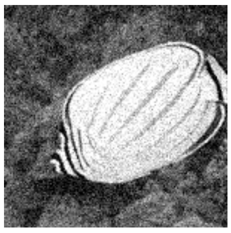
(a) Original $\mathbf{u}^0$

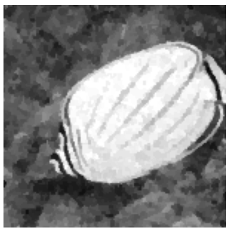

(b) Noisy image $\mathbf{u}^\delta$


(c) Restored image $\mathbf{u}$


(d) Error $|\mathbf{u}^0 - \mathbf{u}|$

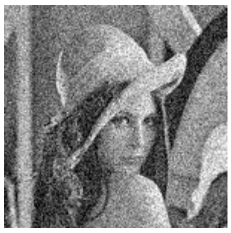

(e) Original $\mathbf{u}^0$

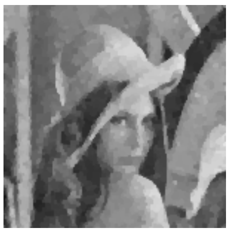

(f) Noisy image $\mathbf{u}^\delta$


(g) Restored image $\mathbf{u}$

(h) Error $|\mathbf{u}^0 - \mathbf{u}|$

(i) Original $\mathbf{u}^0$


(j) Noisy image $\mathbf{u}^\delta$


(k) Restored image $\mathbf{u}$

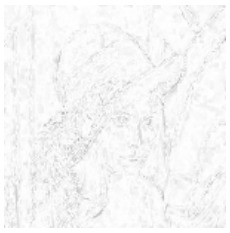
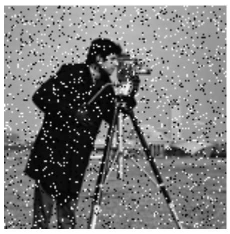
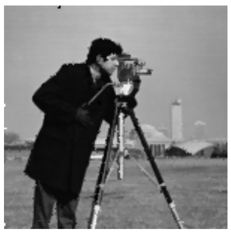

(l) Error $|\mathbf{u}^0 - \mathbf{u}|$

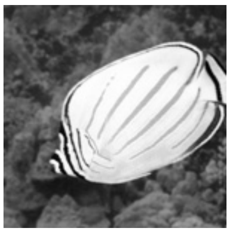
Figure 5.5: Gaussian degraded images restored by the anisotropic TV model.

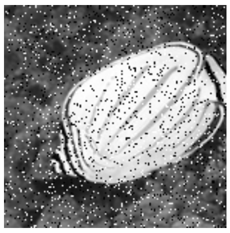
(a) Original $\mathbf{u}^0$

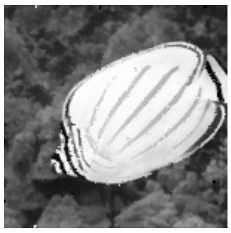

(b) Noisy image $\mathbf{u}^\delta$

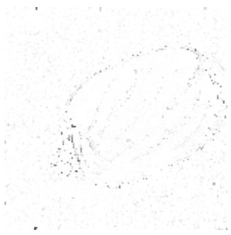

(c) Restored image $\mathbf{u}$

(d) Error $|\mathbf{u}^0 - \mathbf{u}|$

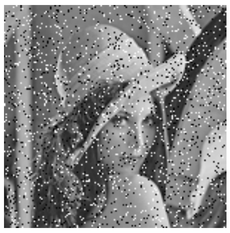

(e) Original $\mathbf{u}^0$

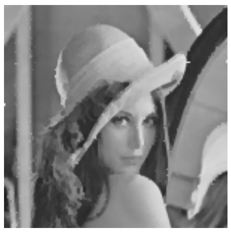

(f) Noisy image $\mathbf{u}^\delta$


(g) Restored image $\mathbf{u}$


(h) Error $|\mathbf{u}^0 - \mathbf{u}|$

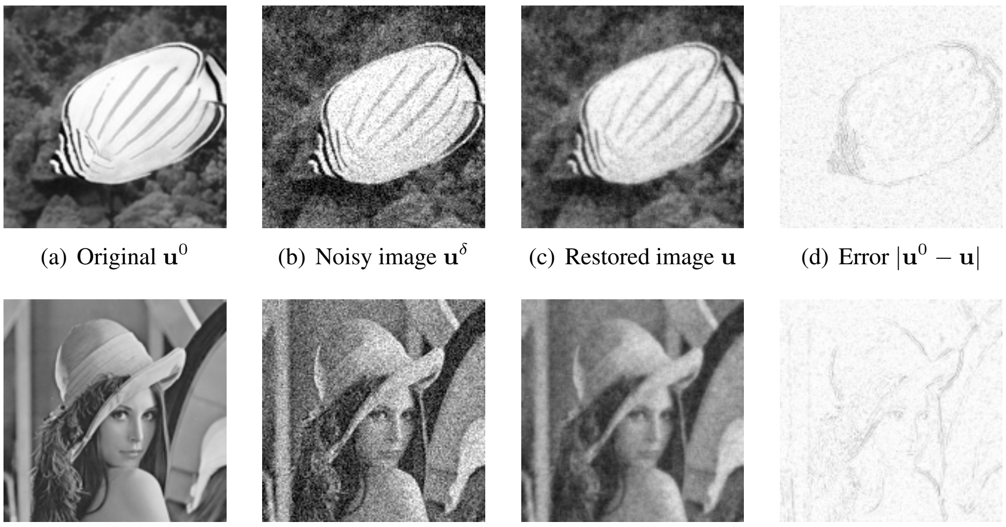
(i) Original $\mathbf{u}^0$


(j) Noisy image $\mathbf{u}^\delta$


(k) Restored image $\mathbf{u}$

(l) Error $|\mathbf{u}^0 - \mathbf{u}|$

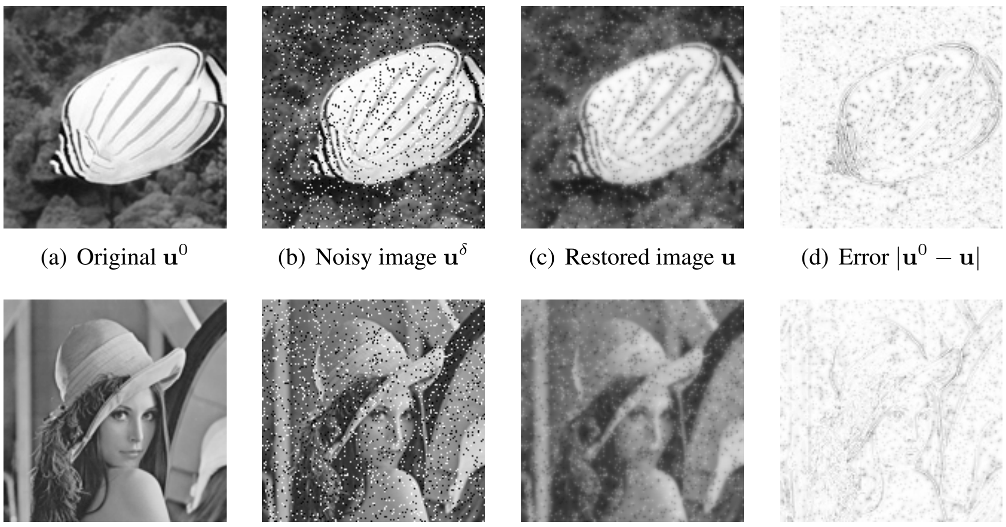
Figure 5.6: Salt & Pepper degraded images restored by the anisotropic TV model.



(a) Original $\mathbf{u}^0$ (b) Noisy image $\mathbf{u}^\delta$ (c) Restored image $\mathbf{u}$ (d) Error $|\mathbf{u}^0 - \mathbf{u}|$

(e) Original $\mathbf{u}^0$ (f) Noisy image $\mathbf{u}^\delta$ (g) Restored image $\mathbf{u}$ (h) Error $|\mathbf{u}^0 - \mathbf{u}|$

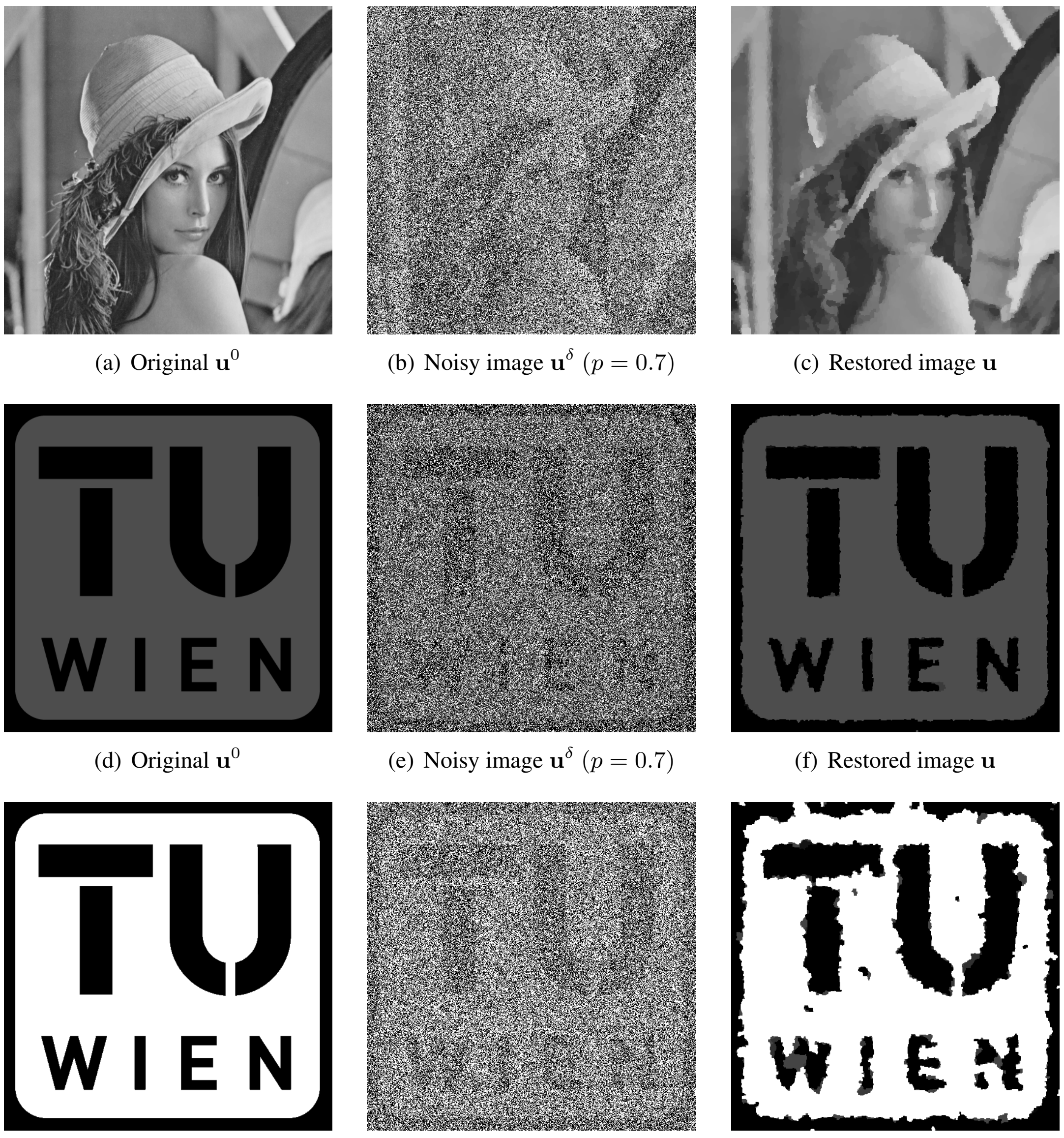
Figure 5.7: Gaussian degraded images restored by the isotropic model.



(a) Original $\mathbf{u}^0$ (b) Noisy image $\mathbf{u}^\delta$ (c) Restored image $\mathbf{u}$ (d) Error $|\mathbf{u}^0 - \mathbf{u}|$

(e) Original $\mathbf{u}^0$ (f) Noisy image $\mathbf{u}^\delta$ (g) Restored image $\mathbf{u}$ (h) Error $|\mathbf{u}^0 - \mathbf{u}|$

Figure 5.8: Salt & Pepper degraded images restored by the isotropic model.



(a) Original $\mathbf{u}^0$ (b) Noisy image $\mathbf{u}^\delta$ $(p = 0.7)$ (c) Restored image $\mathbf{u}$

(d) Original $\mathbf{u}^0$ (e) Noisy image $\mathbf{u}^\delta$ $(p = 0.7)$ (f) Restored image $\mathbf{u}$

(g) Original $\mathbf{u}^0$ (h) Noisy image $\mathbf{u}^\delta$ $(p = 0.9)$ (i) Restored image $\mathbf{u}$

Figure 5.9: Heavily distorted images reconstructed with the anisotropic TV model.

CHAPTER 6

Conclusion and Future Work

In this thesis we have investigated the applicability of graph cut methods to continuous (convex) first-order regularization functionals which are frequently used for the task of image denoising.

In Chapter 3 we have introduced stochastic models for image modeling and in particular Markov random fields. Moreover, we presented a Bayesian justification for energy minimization in order to infer the maximum a posterior estimate. As soon as an image is modeled as an MRF, powerful inferences can be made. For instance, in image denoising one wants to find the true intensity having observed some noisy image which might have been corrupted in the process of image acquisition. The MAP estimate then gives the most likely explanation for the observed data and allows the inference of a restored image. The main concept of this chapter was the insight that by minimizing some energy function directly leads to the MAP estimate.

Moreover, in Chapter 4 we presented graph cut-based methods for energy minimization. We discussed the direct correspondence between (quadratic) pseudo-Boolean functions and the cost function of a graph cut in a flow network. Furthermore, we showed that such a (directed) cut function is always submodular and thus a (quadratic) pseudo-Boolean function can be minimized exactly with a graph cut if and only if it is submodular. We have seen how the important max-flow min-cut theorem then allows the efficient computation of a minimum cut. As a central result of this chapter, we showed how to relate MRF energies (and thus posterior probabilities) to graph cut functions.

However, as soon as the energies increase in complexity either by violating submodularity or by extending the range of labels (multilabel problem), the problem renders NP-hard. We gave a proof for the NP-completeness of even very simple nonsubmodular Boolean MRF energies and thus making approximations inevitable. For the multilabel problem we discussed so called "move-making" algorithms: the expansion and the swap algorithm. In case the MRF energy is nonsubmodular the fusion move can still be applied.

After establishing the basics, we addressed continuous first-order regularization functionals for image denoising. For a subset of the functionals summarized by Scherzer et al. [103] we investigated the applicability of graph cut-based methods. In particular, we have investigated a discrete form of the anisotropic regularization model. As a theoretical result we found that our discrete form generalizes isotropic regularization and in general results in nonsubmodular energies. Moreover, we showed that such models can neither be approximated with the expansion nor with the swap algorithm.

The experimental contribution of this thesis are graph constructions for the expansion, the swap, and the fusion algorithm and moreover, we conducted several experiments. Our results give evidence that even though the anisotropic regularization functionals can only be approximated, the fusion algorithm yields acceptable results for Gaussian degraded images and is capable of preserving object boundaries. Another advantage of this method is the fast convergence.

Nevertheless, the fact that only few interesting MRF energies can be minimized exactly in polynomial time is very unsatisfying. Possible future work includes the investigation of other, more complex graph problems such as the minimum-cost flow problem or as a variant the minimum-cost maximum flow problem, for which polynomial-time algorithms exist. Moreover, the multi-commodity flow problem might be a suitable candidate for generalization.

Regarding the discussed discrete models future work may include the consideration of 8-connected or 16-connected neighborhoods to incorporate higher-order energy potentials. Moreover, our result regarding the discrete anisotropic model is very general and should be subject to further investigation. Finally, the effect of the proposal generation in the fusion algorithm should be studied.

Bibliography

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive Digital Photomontage. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 23(3):294–302, 2004.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] C. Arora, S. Banerjee, P. Kalra, and S. Maheshwari. An Efficient Graph Cut Algorithm for Computer Vision Problems. In *Computer Vision – ECCV 2010*, volume 6313, pages 552–565. Springer, 2010.

[4] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer, first edit edition, 2001.

[5] E. Bae, J. Shi, and X.-C. Tai. Graph cuts for curvature based image denoising. *IEEE Transactions on Image Processing*, 20(5):1199–1210, May 2011.

[6] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

[7] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.

[8] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 489–495. IEEE, 1999.

[9] A. Blake, C. Rother, M. Brown, P. Perez, and P. H. S. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 428–441. Springer, 2004.

[10] A. Blake, P. Kohli, and C. Rother. *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.

[11] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[12] E. Boros, P. L. Hammer, and X. Sun. Network Flows and Minimization of Quadratic Pseudo-Boolean Functions. Technical report, Rutgers Center for Operations Research, 1991.

[13] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[14] Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In S. Delp, A. DiGoia, and B. Jaramaz, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2000*, volume 1935 of *Lecture Notes in Computer Science*, pages 147–175. Springer, 2000.

[15] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

[16] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision. In *EMMCVPR*, pages 359–374, 2001.

[17] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proceedings Ninth IEEE International Conference on Computer Vision*, number Iccv, pages 26–33 vol.1. IEEE Computer Society, 2003.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[19] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 648–655. IEEE, 1998.

[20] Y. Boykov, O. Veksler, and R. Zabih. A New Algorithm for Energy Minimization with Discontinuities Energy Minimization in Early Vision. In *EMMCVPR*, pages 205–220, 1999.

[21] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. In *Proceedings of the Seventh IEEE International Conference on Computer Vision. Kerkyra, Greece*, volume 1, pages 377–384. IEEE Computer Society, 1999.

[22] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[23] A. Chambolle and J. Darbon. On Total Variation Minimization and Surface Evolution Using Parametric Maximum Flows. *International Journal of Computer Vision*, 84(3):288–307, Apr. 2009.

[24] R. Chan, T. F. Chan, and A. Yip. Numerical Methods and Applications in Total Variation Image Restoration. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, chapter 24, pages 1059–1094. Springer, New York, 1 edition, 2011.

[25] G. Charpiat. Exhaustive Family of Energies Minimizable Exactly by a Graph Cut. In *CVPR*, page 8, 2011.

[26] T. Cooke. Two Applications of Graph-Cuts to Image Processing. In *2008 Digital Image Computing: Techniques and Applications*, pages 498–504. Ieee, 2008.

[27] D. Cremers and L. Grady. Statistical Priors for Efficient Combinatorial Optimization Via Graph Cuts. In *ECCV (3)*, pages 263–274, 2006.

[28] W. H. Cunningham. Minimum cuts, modular functions, and matroid polyhedra. *Networks*, 15(2):205–215, Jan. 1985.

[29] J. Darbon. Total variation minimization with L1 data fidelity as a contrast invariant filter. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, number 1, pages 221–226. Ieee, 2005.

[30] J. Darbon. Global optimization for first order Markov Random Fields with submodular priors. *Discrete Applied Mathematics*, 157(16):3412–3423, Aug. 2009.

[31] J. Darbon and S. Peyronnet. A Vectorial Self-dual Morphological Filter Based on Total Variation Minimization. In G. Bebis, R. Boyle, D. Koracin, and B. Parvin, editors, *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 388–395. Springer, 2005.

[32] J. Darbon and M. Sigelle. Exact Optimization of Discrete Constrained Total Variation Minimization Problems. In R. Klette and J. Žunić, editors, *Combinatorial Image Analysis*, volume 3322 of *Lecture Notes in Computer Science*, pages 548–557. Springer, Berlin / Heidelberg, 2004.

[33] J. Darbon and M. Sigelle. A Fast and Exact Algorithm for Total Variation Minimization. In J. Marques, N. Pérez de la Blanca, and P. Pina, editors, *Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science*, pages 351–359. Springer, Berlin / Heidelberg, 2005.

[34] J. Darbon and M. Sigelle. Image Restoration with Discrete Constrained Total Variation Part I: Fast and Exact Optimization. *Journal of Mathematical Imaging and Vision*, 26(3): 261–276, Aug. 2006.

[35] J. Darbon and M. Sigelle. Image Restoration with Discrete Constrained Total Variation Part II: Levelable Functions, Convex Priors and Non-Convex Cases. *Journal of Mathematical Imaging and Vision*, 26(3):277–291, Nov. 2006.

[36] A. Delong and Y. Boykov. A Scalable Graph-Cut Algorithm for N-D grids. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8. IEEE, June 2008.

[37] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *CVPR*, pages 2173–2180. IEEE, 2010.

[38] N. Y. El-Zehiry and A. Elmaghraby. A graph cut based active contour for multiphase image segmentation. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, number 2, pages 3188–3191. IEEE, 2008.

[39] N. Y. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the Mumford-Shah model. In *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*, pages 182–187. ACTA Press, 2007.

[40] P. Elias, A. Feinstein, and C. E. Shannon. A note on the maximum flow through a network. *IRE Transactions on Information Theory*, 2(4):117–119, Dec. 1956.

[41] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

[42] D. Freedman and P. Drineas. Energy Minimization via Graph Cuts: Settling What is Possible. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 939–946. IEEE, 2005.

[43] D. Freedman and M. W. Turek. Illumination-invariant tracking via graph cuts. In *CVPR*, pages 10 – 17 vol. 2. IEEE Computer Society, 2005.

[44] D. Freedman and M. W. Turek. Graph cuts with many-pixel interactions: Theory and applications to shape modelling. *Image and Vision Computing*, 28(3):467–473, Mar. 2010.

[45] S. Fujishige. *Submodular Functions and Optimization*. Annals of Discrete Mathematics. Elsevier, second edi edition, 2005.

[46] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[47] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(9):721–741, 1984.

[48] A. V. Goldberg. Two-Level Push-Relabel Algorithm for the Maximum Flow Problem. In *Algorithmic Aspects in Information and Management*, Lecture Notes in Computer Science, pages 212–225. Springer Berlin Heidelberg, 2009.

[49] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, Oct. 1988.

[50] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51 (2):271–279, 1989.

[51] G. Hamarneh. Multi-label MRF Optimization via a Least Squares s- t Cut. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I*, pages 1055–1066. Springer, 2009.

[52] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Programming*, 28(2):121–155, 1984.

[53] J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Technical report, University of California, Berkeley, 1968.

[54] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov random fields and related problems. *Journal of the ACM*, 48(4):686–701, July 2001.

[55] P. J. Huber. *Robust Statistics*, volume 1 of *Wiley Series in Probability and Statistics*. Wiley & Sons, 1981.

[56] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, Oct. 2003.

[57] H. Ishikawa. Transformation of General Binary MRF Minimization to the First-Order Case. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1234–1249, 2011.

[58] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *Computer Vision and Pattern Recognition, 1998.*, pages 125–131. IEEE Computer Society, 1998.

[59] H. Ishikawa and D. Geiger. Occlusions, Discontinuities, and Epipolar Lines in Stereo. In *ECCV*, number ii, pages 232–248, 1998.

[60] H. Ishikawa and D. Geiger. Mapping Image Restoration to a Graph Problem. In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99)*, pages 189–193, 1999.

[61] P. L. Ivănescu. Some Network Flow Problems Solved with Pseudo-Boolean Programming. *Operations Research*, 13(3):388–399, 1965.

[62] S. Iwata, L. Fleischer, and S. Fujishige. A Combinatorial Strongly Polynomial Algorithm for Minimizing Submodular Functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[63] S. Jegelka and J. Bilmes. Submodularity Beyond Submodular Energies: Coupling Edges in Graph Cuts. In *Computer Vision and Pattern Recognition (CVPR 2011)*, page 8, 2011.

[64] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proceedings Ninth IEEE International Conference on Computer Vision (ICCV03)*, volume 2, pages 1033–1040. IEEE, 2003.

[65] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 14–23. IEEE Comput. Soc, 1999.

[66] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639, Sept. 2002.

[67] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2005.

[68] P. Kohli. *Minimizing dynamic and higher order energy functions using graph cuts*. PhD thesis, Oxford Brookes University, 2007.

[69] P. Kohli and P. H. S. Torr. Effciently solving dynamic markov random fields using graph cuts. In *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, pages 922–929. IEEE Computer Society, 2005.

[70] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2079–88, Dec. 2007.

[71] V. Kolmogorov. *Graph based algorithms for scene reconstruction from two or more views*. PhD thesis, Cornell University, 2004.

[72] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[73] V. Kolmogorov and R. Zabih. Computing Visual Correspondence with Occlusions using Graph Cuts. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 508–515. IEEE Comput. Soc, 2001.

[74] V. Kolmogorov and R. Zabih. Multi-camera Scene Reconstruction via Graph Cuts. In *ECCV*, pages 8–40, 2002.

[75] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? In *ECCV*, pages 65–81, 2002.

[76] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):65–81, 2004.

[77] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Probabilistic fusion of stereo with color and contrast for bilayer segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(9):1480–1492, Sept. 2006.

[78] H. R. Künsch. Robust priors for smoothing and image restoration. *Annals of the Institute of Statistical Mathematics*, 46(1):1–19, 1994.

[79] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 1: 10, 2003.

[80] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*, pages 282–289, 2001.

[81] V. Lempitsky, C. Rother, and A. Blake. LogCut - Efficient Graph Cut Optimization for Markov Random Fields. In *ICCV*, pages 1–8, 2007.

[82] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, pages 1–8. Ieee, June 2008.

[83] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion Moves for Markov Random Field Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8): 1392–1405, Aug. 2010.

[84] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 3rd edition, 2009.

[85] F. Liers and G. Pardella. Simplifying maximum flow computations: The effect of shrinking and good initial flows. *Discrete Applied Mathematics*, 159(17):2187–2203, Oct. 2011.

[86] L. Lovász. Submodular functions and convexity. In A. Bachem, M. Grötschel, and B. H. Korte, editors, *Mathematical Programming - the state of the art*, pages 235–257. Springer, 1983.

[87] K. Murota. *Discrete Convex Analysis*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2003.

[88] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions. *Mathematical Programming*, 14:265–294, 1978.

[89] J. B. Orlin. A Faster Strongly Polynomial Time Algorithm for Submodular Function Minimization. *Mathematical Programming*, 118(2):237–251, Nov. 2009.

[90] J.-C. Picard and H. D. Ratliff. A Graph-Theoretic Equivalence for Integer Programs. *Operations Research*, 21(1):261–269, Jan. 1973.

[91] J.-C. Picard and H. D. Ratliff. Minimum cuts and related problems. *Networks*, 5(4): 357–370, 1975.

[92] R. B. Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(01):106–109, 1952.

[93] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[94] C. H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 83(10):177–183, 1967.

[95] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, Aug. 2004.

[96] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *CVPR*, volume 1, pages 589–596, 2005.

[97] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 993–1000. IEEE, 2006.

[98] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing Binary MRFs via Extended Roof Duality. Technical report msr-tr-2007-46, Microsoft Research, 2007.

[99] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing Binary MRFs via Extended Roof Duality. In *CVPR*, pages 1–8. IEEE, June 2007.

[100] S. Roy and I. J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In *Sixth International Conference on Computer Vision*, pages 492–499. Narosa Publishing House, 1998.

[101] L. I. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, Nov. 1992.

[102] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[103] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational Methods in Imaging*. Springer, 1 edition, 2009.

[104] D. Schlesinger and B. Flach. Transforming an arbitrary minsum problem into a binary one. Technical report, Dresden University of Technology, Apr. 2006.

[105] I. J. Schoenberg. Spline Functions and the Problem of Graduation. *Proceedings of the National Academy of Sciences of the United States of America*, 52(4):947–950, 1964.

[106] A. Schrijver. *Combinatorial Optimization*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[107] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[108] H. S. Stone. Multiprocessor Scheduling with the Aid of Network Flow Algorithms. *IEEE Transactions on Software Engineering*, SE-3(1):85–93, Jan. 1977.

[109] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[110] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. Winston & Sons, Washington, 1977.

[111] O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.

[112] O. Veksler. Image segmentation by nested cuts. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 339–344. IEEE, 2000.

[113] O. Veksler. Graph Cut Based Optimization for MRFs with Truncated Convex Priors. In *CVPR*, pages 1–8, 2007.

[114] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1–8. IEEE, June 2008.

[115] J. Wills, S. Agarwal, and S. Belongie. What Went Where. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, page 8, 2003.

[116] G. Winkler. *Image Analysis, Random Fields and Markov Chan Monte Carlo Methods*, volume 27 of *Applications of Methematics*. Springer, second edi edition, 2003.

[117] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2008.

[118] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *CVPR*, pages 437–444, 2004.

[119] B. Zalesky. Network flow optimization for restoration of images. *Journal of Applied Mathematics*, 2(4):199–218, 2002.

List of Figures

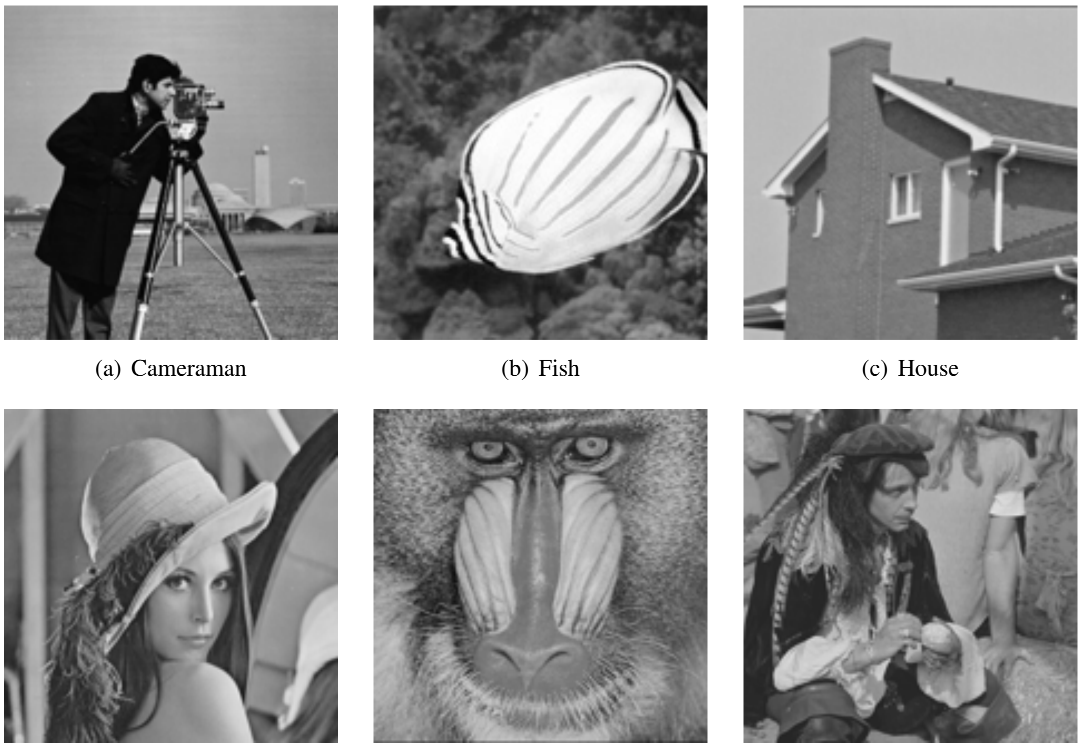
List of Tables

List of Algorithms

APPENDIX A

Results

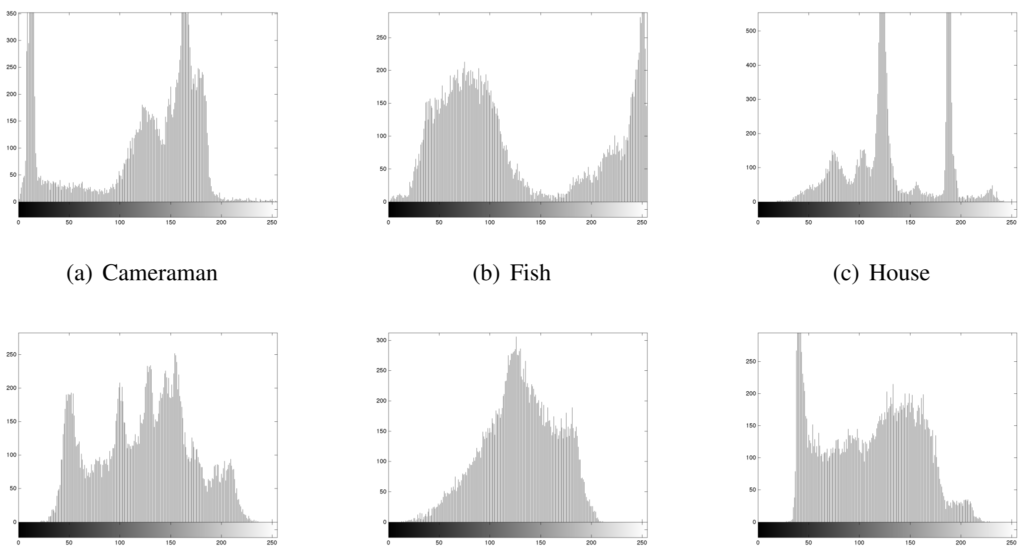
A.1 Test Images



(a) Cameraman (b) Fish (c) House

(d) Lenna (e) Mandrill (f) Pirate

Figure A.1: Test images used.



(a) Cameraman (b) Fish (c) House

(d) Lenna (e) Mandrill (f) Pirate

Figure A.2: Histograms of the undistorted test images.

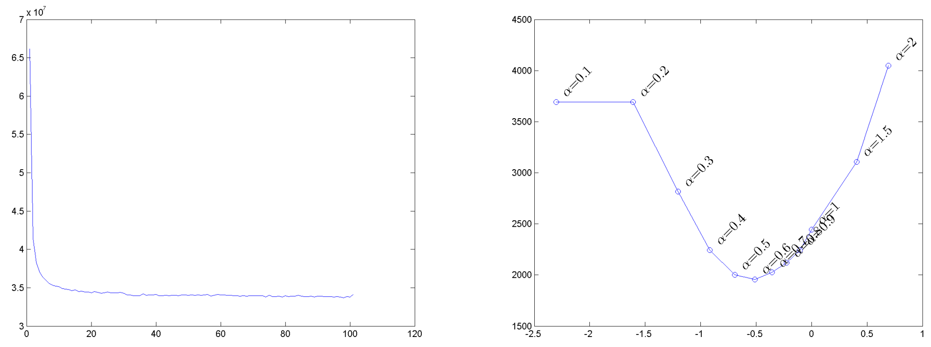
Instance	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_1$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_2$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_1 / len(\mathbf{u}^0)$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_2 / len(\mathbf{u}^0)$	r^δ	$\overline{r}^\delta$
Cameraman	19.4950	0.1641	0.0009	0.0000	1720.1490	0.0765
Fish	18.8812	0.1598	0.0008	0.0000	1665.9882	0.0740
House	20.3844	0.1699	0.0009	0.0000	1798.6275	0.0799
Lenna	20.3170	0.1688	0.0009	0.0000	1792.6745	0.0797
Mandrill	20.4673	0.1703	0.0009	0.0000	1805.9373	0.0803
Pirate	20.3160	0.1695	0.0009	0.0000	1792.5843	0.0797

Table A.1: Characteristics of the degraded instances (additive Gaussian white noise, $\sigma^2 = 0.01$).

A.2 Results

Instance	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_1$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_2$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_{1}/len(\mathbf{v})$	$\|\mathbf{u}^0 - \mathbf{u}^\delta\|_{2}/len(\mathbf{v})$	r^δ	$\overline{r}^\delta$
Cameraman	12.7663	0.2981	0.0006	0.0000	1126.4392	0.0501
Fish	12.6471	0.3143	0.0006	0.0000	1115.9216	0.0496
House	13.0100	0.2875	0.0006	0.0000	1147.9451	0.0510
Lenna	13.0480	0.2915	0.0006	0.0000	1151.2941	0.0512
Mandrill	12.9640	0.2818	0.0006	0.0000	1143.8784	0.0508
Pirate	12.3627	0.2830	0.0005	0.0000	1090.8275	0.0485

Table A.2: Characteristics of the degraded instances (Salt & Pepper noise, $p = 0.1$).



(a) Typical fusion move energy (x-axis: iteration, y-axis: energy).

(b) L^2 error vs. $\log(\alpha)$ plot of the results for the Cameraman instance.

Figure A.3: Energy curves.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
438629.0000	3691.9265	19.4946	0.1641	1720.1137	0.0764	0.0001	1	20	107.5383	3837.5706	3870.5221	1.0086
438143.0000	3691.9991	19.4730	0.1641	1718.2078	0.0764	0.0010	1	20	108.5458	38375.7061	44915.4056	1.1704
385732.0000	3322.6486	17.1436	0.1477	1512.6745	0.0672	0.0100	1	20	109.2122	383757.0611	366123.6747	0.9541
280803.0000	2576.9821	12.4801	0.1145	1101.1882	0.0489	0.1000	1	20	110.4075	3837570.6107	922160.3760	0.2403
273156.0000	2543.9568	12.1403	0.1131	1071.2000	0.0476	1.0000	1	20	110.5279	38375706.1067	4234483.4231	0.1103
274425.0000	2573.8526	12.1967	0.1144	1076.1765	0.0478	10.0000	1	20	109.6476	383757061.0674	38266462.9264	0.0997
272930.0000	2555.8775	12.1302	0.1136	1070.3137	0.0476	100.0000	1	20	109.7310	3837570610.6741	371147793.3664	0.0967
273884.0000	2575.3516	12.1726	0.1145	1074.0549	0.0477	1000.0000	1	20	108.5975	38375706106.7407	3657527913.8385	0.0953
272089.0000	2547.9343	12.0928	0.1132	1067.0157	0.0474	10000.0000	1	20	109.1731	383757061067.4069	37270704348.8963	0.0971

Table A.3: Cameraman, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
438638.0000	3691.9772	19.4950	0.1641	1720.1490	0.0765	0.0001	1	20	123.9819	71.6287	71.6287	1.0000
438638.0000	3691.9772	19.4950	0.1641	1720.1490	0.0765	0.0010	1	20	121.9590	716.2867	716.2867	1.0000
438633.0000	3691.9449	19.4948	0.1641	1720.1294	0.0765	0.0100	1	20	122.7158	7162.8666	7213.7960	1.0071
437445.0000	3687.3629	19.4420	0.1639	1715.4706	0.0762	0.1000	1	20	122.3083	71628.6662	80181.1020	1.1194
312030.0000	2805.8279	13.8680	0.1247	1223.6471	0.0544	1.0000	1	20	122.3148	716286.6618	644405.5080	0.8996
268584.0000	2515.0734	11.9371	0.1118	1053.2706	0.0468	10.0000	1	20	122.7648	7162866.6183	2351688.4063	0.3283
270862.0000	2539.7106	12.0383	0.1129	1062.2039	0.0472	100.0000	1	20	123.3498	71628666.1835	19088321.0868	0.2665
272304.0000	2547.0866	12.1024	0.1132	1067.8588	0.0475	1000.0000	1	20	123.0496	716286661.8349	186493955.5118	0.2604
270719.0000	2538.2764	12.0320	0.1128	1061.6431	0.0472	10000.0000	1	20	123.5926	7162866618.3489	1865874839.7523	0.2605

Table A.4: Cameraman, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
438634.0000	3691.9884	19.4948	0.1641	1720.1333	0.0765	0.0001	1	20	108.4954	3837.5706	3867.6460	1.0078
438543.0000	3691.6732	19.4908	0.1641	1719.7765	0.0764	0.0010	1	20	107.4626	38375.7061	41038.8679	1.0694
436134.0000	3679.9788	19.3837	0.1636	1710.3294	0.0760	0.0100	1	20	107.7512	383757.0611	438510.8416	1.1427
412545.0000	3556.6733	18.3353	0.1581	1617.8235	0.0719	0.1000	1	20	107.7899	3837570.6107	4459312.2867	1.1620
307058.0000	2747.8293	13.6470	0.1221	1204.1490	0.0535	1.0000	1	20	108.7797	38375706.1067	18724746.0295	0.4879
268813.0000	2498.7203	11.9472	0.1111	1054.1686	0.0469	10.0000	1	20	108.9332	383757061.0674	56701307.5478	0.1478
272083.0000	2544.0717	12.0926	0.1131	1066.9922	0.0474	100.0000	1	20	108.8027	3837570610.6741	397018116.1888	0.1035
276158.0000	2569.1890	12.2737	0.1142	1082.9725	0.0481	1000.0000	1	20	108.9882	38375706106.7407	3778684171.8221	0.0985
274107.0000	2573.1080	12.1825	0.1144	1074.9294	0.0478	10000.0000	1	20	108.9139	383757061067.4069	37722140666.7929	0.0983

Table A.5: Cameraman, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
438638.0000	3691.9772	19.4950	0.1641	1720.1490	0.0765	0.0001	1	20	120.9994	71.6287	71.6287	1.0000
438638.0000	3691.9772	19.4950	0.1641	1720.1490	0.0765	0.0010	1	20	122.6056	716.2867	716.2867	1.0000
438638.0000	3691.9686	19.4950	0.1641	1720.1490	0.0765	0.0100	1	20	121.8297	7162.8666	7190.8320	1.0039
438461.0000	3691.2539	19.4872	0.1641	1719.4549	0.0764	0.1000	1	20	122.2931	71628.6662	75581.2488	1.0552
434798.0000	3674.5732	19.3244	0.1633	1705.0902	0.0758	1.0000	1	20	121.9359	716286.6618	788758.6947	1.1012
391868.0000	3452.0521	17.4164	0.1534	1536.7373	0.0683	10.0000	1	20	122.4615	7162866.6183	7879774.8890	1.1001
267111.0000	2440.8890	11.8716	0.1085	1047.4941	0.0466	100.0000	1	20	123.5380	71628666.1835	35135173.0084	0.4905
267063.0000	2509.3499	11.8695	0.1115	1047.3059	0.0465	1000.0000	1	20	123.5830	716286661.8349	201783340.4068	0.2817
270681.0000	2536.6693	12.0303	0.1127	1061.4941	0.0472	10000.0000	1	20	124.0356	7162866618.3489	1882315294.8022	0.2628

Table A.6: Cameraman, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
424831.0000	3594.4870	18.8814	0.1598	1666.0039	0.0740	0.0001	1	20	109.0656	3412.1930	3434.0426	1.0064
424733.0000	3597.2549	18.8770	0.1599	1665.6196	0.0740	0.0010	1	20	108.6568	34121.9302	39437.8284	1.1558
382369.0000	3318.4200	16.9942	0.1475	1499.4863	0.0666	0.0100	1	20	108.6748	341219.3022	335423.9418	0.9830
285852.0000	2628.1933	12.7045	0.1168	1120.9882	0.0498	0.1000	1	20	109.6384	3412193.0222	918852.3447	0.2693
289150.0000	2634.1887	12.8511	0.1171	1133.9216	0.0504	1.0000	1	20	110.4961	34121930.2216	4366437.9097	0.1280
292507.0000	2651.2173	13.0003	0.1178	1147.0863	0.0510	10.0000	1	20	109.7455	341219302.2159	38519975.0928	0.1129
295676.0000	2662.0755	13.1412	0.1183	1159.5137	0.0515	100.0000	1	20	110.1337	3412193022.1585	382071290.9581	0.1120
294040.0000	2665.7367	13.0684	0.1185	1153.0980	0.0512	1000.0000	1	20	110.8086	34121930221.5852	3869708122.6699	0.1134
292215.0000	2640.0244	12.9873	0.1173	1145.9412	0.0509	10000.0000	1	20	109.5214	341219302215.8519	38297452105.9897	0.1122

Table A.7: Fish, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
424827.0000	3594.5037	18.8812	0.1598	1665.9882	0.0740	0.0001	1	20	121.7291	66.5416	66.5416	1.0000
424827.0000	3594.5037	18.8812	0.1598	1665.9882	0.0740	0.0010	1	20	123.4495	665.4160	665.4160	1.0000
424827.0000	3594.5076	18.8812	0.1598	1665.9882	0.0740	0.0100	1	20	122.1081	6654.1600	6696.1302	1.0063
422986.0000	3587.3355	18.7994	0.1594	1658.7686	0.0737	0.1000	1	20	121.8272	66541.6001	74328.6078	1.1170
320139.0000	2892.0265	14.2284	0.1285	1255.4471	0.0558	1.0000	1	20	122.1055	665416.0008	621207.1176	0.9336
285859.0000	2598.4059	12.7048	0.1155	1121.0157	0.0498	10.0000	1	20	122.3875	6654160.0085	2369796.6677	0.3561
287089.0000	2611.4906	12.7595	0.1161	1125.8392	0.0500	100.0000	1	20	122.5022	66541600.0846	19462032.6302	0.2925
291554.0000	2634.3565	12.9580	0.1171	1143.3490	0.0508	1000.0000	1	20	122.8267	665416000.8462	191150592.1395	0.2873
287943.0000	2612.9545	12.7975	0.1161	1129.1882	0.0502	10000.0000	1	20	122.7365	6654160008.4624	1909120516.7479	0.2869

Table A.8: Fish, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
424823.0000	3594.4931	18.8810	0.1598	1665.9725	0.0740	0.0001	1	20	108.4826	3412.1930	3424.1889	1.0035
424655.0000	3593.7522	18.8736	0.1597	1665.3137	0.0740	0.0010	1	20	108.7429	34121.9302	36341.2513	1.0650
422683.0000	3584.4538	18.7859	0.1593	1657.5804	0.0737	0.0100	1	20	108.3644	341219.3022	386962.2927	1.1341
402207.0000	3480.3309	17.8759	0.1547	1577.2824	0.0701	0.1000	1	20	108.3488	3412193.0222	4003158.0180	1.1732
310476.0000	2796.6605	13.7989	0.1243	1217.5529	0.0541	1.0000	1	20	108.2837	34121930.2216	17630517.2222	0.5167
280319.0000	2554.6790	12.4586	0.1135	1099.2902	0.0489	10.0000	1	20	108.6740	341219302.2159	57002190.2819	0.1671
290016.0000	2628.9873	12.8896	0.1168	1137.3176	0.0505	100.0000	1	20	108.6005	3412193022.1585	406259045.6858	0.1191
294534.0000	2654.0539	13.0904	0.1180	1155.0353	0.0513	1000.0000	1	20	108.5349	34121930221.5852	3849215984.9123	0.1128
294632.0000	2660.3575	13.0948	0.1182	1155.4196	0.0514	10000.0000	1	20	108.6146	341219302215.8519	38198120729.2907	0.1119

Table A.9: Fish, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
424827.0000	3594.5037	18.8812	0.1598	1665.9882	0.0740	0.0001	1	20	121.6740	66.5416	66.5416	1.0000
424827.0000	3594.5037	18.8812	0.1598	1665.9882	0.0740	0.0010	1	20	121.7688	665.4160	665.4160	1.0000
424820.0000	3594.4557	18.8809	0.1598	1665.9608	0.0740	0.0100	1	20	121.9156	6654.1600	6687.0486	1.0049
424680.0000	3593.8072	18.8747	0.1597	1665.4118	0.0740	0.1000	1	20	122.2385	66541.6001	70213.5738	1.0552
420681.0000	3575.1597	18.6969	0.1589	1649.7294	0.0733	1.0000	1	20	122.0490	665416.0008	730657.6051	1.0980
380889.0000	3358.2762	16.9284	0.1493	1493.6824	0.0664	10.0000	1	20	121.8942	6654160.0085	7307822.2982	1.0982
275628.0000	2489.0372	12.2501	0.1106	1080.8941	0.0480	100.0000	1	20	122.3500	66541600.0846	34450451.5913	0.5177
286372.0000	2594.5258	12.7276	0.1153	1123.0275	0.0499	1000.0000	1	20	122.7832	665416000.8462	207362918.5627	0.3116
287697.0000	2609.9354	12.7865	0.1160	1128.2235	0.0501	10000.0000	1	20	122.4465	6654160008.4624	1917876195.7990	0.2882

Table A.10: Fish, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
458651.0000	3822.6730	20.3845	0.1699	1798.6314	0.0799	0.0001	1	20	108.0874	3631.5664	3644.6140	1.0036
457812.0000	3819.5306	20.3472	0.1698	1795.3412	0.0798	0.0010	1	20	108.0999	36315.6638	42562.9341	1.1720
381474.0000	3285.2011	16.9544	0.1460	1495.9765	0.0665	0.0100	1	20	108.8518	363156.6383	355427.6895	0.9787
244775.0000	2154.2676	10.8789	0.0957	959.9020	0.0427	0.1000	1	20	108.6746	3631566.3829	866125.4290	0.2385
226051.0000	2029.3824	10.0467	0.0902	886.4745	0.0394	1.0000	1	20	109.1094	36315663.8292	3861063.5717	0.1063
226012.0000	2034.3132	10.0450	0.0904	886.3216	0.0394	10.0000	1	20	109.2663	363156638.2916	34373298.1251	0.0947
227900.0000	2046.8815	10.1289	0.0910	893.7255	0.0397	100.0000	1	20	109.8587	3631566382.9162	342378218.9618	0.0943
228379.0000	2049.2630	10.1502	0.0911	895.6039	0.0398	1000.0000	1	20	108.9029	36315663829.1624	3407315116.7747	0.0938
226354.0000	2038.2178	10.0602	0.0906	887.6627	0.0395	10000.0000	1	20	108.7684	363156638291.6239	34154486148.9130	0.0940

Table A.11: House, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
458650.0000	3822.6609	20.3844	0.1699	1798.6275	0.0799	0.0001	1	20	121.9107	71.7473	71.7473	1.0000
458650.0000	3822.6609	20.3844	0.1699	1798.6275	0.0799	0.0010	1	20	121.2187	717.4735	717.4735	1.0000
458636.0000	3822.6282	20.3838	0.1699	1798.5725	0.0799	0.0100	1	20	121.2059	7174.7350	7212.6496	1.0053
456983.0000	3816.0030	20.3104	0.1696	1792.0902	0.0796	0.1000	1	20	121.1858	71747.3497	80979.6767	1.1287
297931.0000	2681.1734	13.2414	0.1192	1168.3569	0.0519	1.0000	1	20	121.9997	717473.4974	635617.2418	0.8859
224819.0000	2019.6933	9.9920	0.0898	881.6431	0.0392	10.0000	1	20	122.7310	7174734.9739	2352098.7989	0.3278
223226.0000	2016.3809	9.9212	0.0896	875.3961	0.0389	100.0000	1	20	122.2297	71747349.7394	19234401.2476	0.2681
225160.0000	2018.5222	10.0071	0.0897	882.9804	0.0392	1000.0000	1	20	122.4646	717473497.3935	188692636.2874	0.2630
223122.0000	2016.5823	9.9165	0.0896	874.9882	0.0389	10000.0000	1	20	122.0931	7174734973.9351	1860248111.1292	0.2593

Table A.12: House, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
458656.0000	3822.6891	20.3847	0.1699	1798.6510	0.0799	0.0001	1	20	107.4094	3631.5664	3647.4852	1.0044
458501.0000	3822.1587	20.3778	0.1699	1798.0431	0.0799	0.0010	1	20	107.6873	36315.6638	39085.0558	1.0763
455477.0000	3809.3597	20.2434	0.1693	1786.1843	0.0794	0.0100	1	20	107.9937	363156.6383	417034.9261	1.1484
425329.0000	3658.8617	18.9035	0.1626	1667.9569	0.0741	0.1000	1	20	108.2338	3631566.3829	4214658.7932	1.1606
293768.0000	2616.7686	13.0564	0.1163	1152.0314	0.0512	1.0000	1	20	108.2140	36315663.8292	17243741.3555	0.4748
228764.0000	2029.9365	10.1673	0.0902	897.1137	0.0399	10.0000	1	20	108.8860	363156638.2916	50571400.0524	0.1393
225913.0000	2034.3173	10.0406	0.0904	885.9333	0.0394	100.0000	1	20	109.1350	3631566382.9162	350602122.9622	0.0965
224385.0000	2025.0040	9.9727	0.0900	879.9412	0.0391	1000.0000	1	20	108.8207	36315663829.1624	3345177430.4724	0.0921
225507.0000	2024.7234	10.0225	0.0900	884.3412	0.0393	10000.0000	1	20	108.8446	363156638291.6239	33611016823.4795	0.0926

Table A.13: House, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
458650.0000	3822.6609	20.3844	0.1699	1798.6275	0.0799	0.0001	1	20	121.4617	71.7473	71.7473	1.0000
458650.0000	3822.6609	20.3844	0.1699	1798.6275	0.0799	0.0010	1	20	121.2762	717.4735	717.4735	1.0000
458646.0000	3822.6538	20.3843	0.1699	1798.6118	0.0799	0.0100	1	20	121.4943	7174.7350	7204.8514	1.0042
458291.0000	3821.3475	20.3685	0.1698	1797.2196	0.0799	0.1000	1	20	123.0227	71747.3497	75999.3028	1.0593
453295.0000	3801.1944	20.1464	0.1689	1777.6275	0.0790	1.0000	1	20	123.0860	717473.4974	797119.8758	1.1110
401759.0000	3534.9788	17.8560	0.1571	1575.5255	0.0700	10.0000	1	20	122.9997	7174734.9739	7858175.3552	1.0953
234057.0000	2099.6783	10.4025	0.0933	917.8706	0.0408	100.0000	1	20	122.1177	71747349.7394	34333205.4623	0.4785
224235.0000	2019.5651	9.9660	0.0898	879.3529	0.0391	1000.0000	1	20	122.6855	717473497.3935	203274083.4523	0.2833
225668.0000	2032.2180	10.0297	0.0903	884.9725	0.0393	10000.0000	1	20	122.7419	7174734973.9351	1891905003.7954	0.2637

Table A.14: House, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457131.0000	3798.8239	20.3169	0.1688	1792.6706	0.0797	0.0001	1	20	108.0433	3516.7711	3523.7763	1.0020
456301.0000	3796.0984	20.2800	0.1687	1789.4157	0.0795	0.0010	1	20	108.2315	35167.7110	41057.2598	1.1675
385746.0000	3307.8334	17.1443	0.1470	1512.7294	0.0672	0.0100	1	20	108.2725	351677.1097	351118.0547	0.9984
257952.0000	2292.8131	11.4645	0.1019	1011.5765	0.0450	0.1000	1	20	108.8332	3516771.0971	895555.0581	0.2547
245207.0000	2202.2995	10.8981	0.0979	961.5961	0.0427	1.0000	1	20	109.0239	35167710.9714	4159590.7281	0.1183
250110.0000	2237.2510	11.1160	0.0994	980.8235	0.0436	10.0000	1	20	109.0243	351677109.7143	37101267.7585	0.1055
249613.0000	2236.2641	11.0939	0.0994	978.8745	0.0435	100.0000	1	20	109.6425	3516771097.1430	367718563.2087	0.1046
250343.0000	2234.9101	11.1264	0.0993	981.7373	0.0436	1000.0000	1	20	108.8056	35167710971.4299	3610596336.1696	0.1027
249388.0000	2232.9957	11.0839	0.0992	977.9922	0.0435	10000.0000	1	20	109.0761	351677109714.2991	35843744318.0058	0.1019

Table A.15: Lenna, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457132.0000	3798.8275	20.3170	0.1688	1792.6745	0.0797	0.0001	1	20	121.6510	70.4689	70.4689	1.0000
457132.0000	3798.8275	20.3170	0.1688	1792.6745	0.0797	0.0010	1	20	121.0228	704.6886	704.6886	1.0000
457140.0000	3798.8888	20.3173	0.1688	1792.7059	0.0797	0.0100	1	20	121.5613	7046.8864	7080.6851	1.0048
455236.0000	3791.0141	20.2327	0.1685	1785.2392	0.0793	0.1000	1	20	121.8146	70468.8642	79462.3351	1.1276
308352.0000	2753.2279	13.7045	0.1224	1209.2235	0.0537	1.0000	1	20	122.1143	704688.6423	638417.0647	0.9060
245141.0000	2203.8995	10.8952	0.0980	961.3373	0.0427	10.0000	1	20	122.3234	7046886.4232	2458712.2740	0.3489
245506.0000	2209.4882	10.9114	0.0982	962.7686	0.0428	100.0000	1	20	122.6958	70468864.2319	20126712.5411	0.2856
246088.0000	2215.7116	10.9372	0.0985	965.0510	0.0429	1000.0000	1	20	122.9906	704688642.3190	197182958.3845	0.2798
248526.0000	2230.1776	11.0456	0.0991	974.6118	0.0433	10000.0000	1	20	122.7331	7046886423.1904	1976151027.0289	0.2804

Table A.16: Lenna, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457133.0000	3798.8265	20.3170	0.1688	1792.6784	0.0797	0.0001	1	20	108.7981	3516.7711	3527.8219	1.0031
457029.0000	3798.4130	20.3124	0.1688	1792.2706	0.0797	0.0010	1	20	108.6397	35167.7110	37702.2226	1.0721
454022.0000	3785.6820	20.1788	0.1683	1780.4784	0.0791	0.0100	1	20	108.6527	351677.1097	404421.3373	1.1500
424447.0000	3633.9060	18.8643	0.1615	1664.4980	0.0740	0.1000	1	20	109.9328	3516771.0971	4094885.2372	1.1644
301409.0000	2674.5035	13.3960	0.1189	1181.9961	0.0525	1.0000	1	20	109.5584	35167710.9714	17544603.7925	0.4989
246462.0000	2195.8456	10.9539	0.0976	966.5176	0.0430	10.0000	1	20	109.5436	351677109.7143	55067522.0822	0.1566
249537.0000	2231.4975	11.0905	0.0992	978.5765	0.0435	100.0000	1	20	109.7882	3516771097.1430	374695596.6062	0.1065
250930.0000	2244.8185	11.1524	0.0998	984.0392	0.0437	1000.0000	1	20	110.2939	35167710971.4299	3650592278.7801	0.1038
248488.0000	2230.3314	11.0439	0.0991	974.4627	0.0433	10000.0000	1	20	109.6582	351677109714.2991	36820698449.3159	0.1047

Table A.17: Lenna, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457132.0000	3798.8275	20.3170	0.1688	1792.6745	0.0797	0.0001	1	20	122.1588	70.4689	70.4689	1.0000
457132.0000	3798.8275	20.3170	0.1688	1792.6745	0.0797	0.0010	1	20	122.6968	704.6886	704.6886	1.0000
457137.0000	3798.8489	20.3172	0.1688	1792.6941	0.0797	0.0100	1	20	121.9999	7046.8864	7071.8694	1.0035
456941.0000	3798.1534	20.3085	0.1688	1791.9255	0.0796	0.1000	1	20	122.9343	70468.8642	74731.7101	1.0605
452201.0000	3777.1017	20.0978	0.1679	1773.3373	0.0788	1.0000	1	20	123.3264	704688.6423	780707.1887	1.1079
401433.0000	3511.2204	17.8415	0.1561	1574.2471	0.0700	10.0000	1	20	123.2364	7046886.4232	7745983.1224	1.0992
246969.0000	2216.3035	10.9764	0.0985	968.5059	0.0430	100.0000	1	20	123.8005	70468864.2319	35225789.6282	0.4999
245833.0000	2204.6571	10.9259	0.0980	964.0510	0.0428	1000.0000	1	20	124.8090	704688642.3190	215114349.8443	0.3053
248823.0000	2229.8016	11.0588	0.0991	975.7765	0.0434	10000.0000	1	20	123.6893	7046886423.1904	1988717350.9528	0.2822

Table A.18: Lenna, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_\mathbf{u}$	$E_\mathbf{u}/E_{\mathbf{u}^\delta}$
460516.0000	3832.4561	20.4674	0.1703	1805.9451	0.0803	0.0001	1	20	108.5637	3787.9004	3801.8641	1.0037
459742.0000	3829.2122	20.4330	0.1702	1802.9098	0.0801	0.0010	1	20	109.1593	37879.0043	43954.4515	1.1604
393543.0000	3350.0524	17.4908	0.1489	1543.3059	0.0686	0.0100	1	20	109.1577	378790.0430	367077.8515	0.9691
285344.0000	2447.2634	12.6820	0.1088	1118.9961	0.0497	0.1000	1	20	109.2876	3787900.4302	852133.8394	0.2250
286752.0000	2452.5721	12.7445	0.1090	1124.5176	0.0500	1.0000	1	20	109.3022	37879004.3017	3279154.9101	0.0866
287461.0000	2466.1365	12.7760	0.1096	1127.2980	0.0501	10.0000	1	20	110.3721	378790043.0167	27449388.9917	0.0725
287077.0000	2462.7316	12.7590	0.1095	1125.7922	0.0500	100.0000	1	20	110.0569	3787900430.1668	265348998.7149	0.0701
286712.0000	2451.3719	12.7428	0.1089	1124.3608	0.0500	1000.0000	1	20	108.8355	37879004301.6680	2655002554.5107	0.0701
288619.0000	2476.6940	12.8275	0.1101	1131.8392	0.0503	10000.0000	1	20	108.7035	378790043016.6798	26985452972.7287	0.0712

Table A.19: Mandrill, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_\mathbf{u}$	$E_\mathbf{u}/E_{\mathbf{u}^\delta}$
460514.0000	3832.4577	20.4673	0.1703	1805.9373	0.0803	0.0001	1	20	121.0025	73.3701	73.3701	1.0000
460514.0000	3832.4577	20.4673	0.1703	1805.9373	0.0803	0.0010	1	20	121.3980	733.7014	733.7014	1.0000
460521.0000	3832.4774	20.4676	0.1703	1805.9647	0.0803	0.0100	1	20	121.4694	7337.0138	7375.8962	1.0053
459178.0000	3826.2316	20.4079	0.1701	1800.6980	0.0800	0.1000	1	20	121.2820	73370.1380	82861.3368	1.1294
326862.0000	2865.2452	14.5272	0.1273	1281.8118	0.0570	1.0000	1	20	122.1047	733701.3797	652978.8984	0.8900
286160.0000	2453.8264	12.7182	0.1091	1122.1961	0.0499	10.0000	1	20	122.5883	7337013.7965	2323412.1476	0.3167
286700.0000	2455.8909	12.7422	0.1092	1124.3137	0.0500	100.0000	1	20	122.6315	73370137.9650	18602478.7870	0.2535
285676.0000	2454.1944	12.6967	0.1091	1120.2980	0.0498	1000.0000	1	20	122.2360	733701379.6503	181880810.1213	0.2479
286310.0000	2455.6751	12.7249	0.1091	1122.7843	0.0499	10000.0000	1	20	122.2136	7337013796.5034	1784994486.7776	0.2433

Table A.20: Mandrill, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_\mathbf{u}$	$E_\mathbf{u}/E_{\mathbf{u}^\delta}$
460511.0000	3832.4328	20.4672	0.1703	1805.9255	0.0803	0.0001	1	20	107.8833	3787.9004	3800.7541	1.0034
460326.0000	3831.7150	20.4589	0.1703	1805.2000	0.0802	0.0010	1	20	107.9235	37879.0043	40537.8145	1.0702
457930.0000	3819.3099	20.3524	0.1697	1795.8039	0.0798	0.0100	1	20	108.1452	378790.0430	433237.8898	1.1437
432753.0000	3682.7016	19.2335	0.1637	1697.0706	0.0754	0.1000	1	20	108.2021	3787900.4302	4402848.0825	1.1623
316833.0000	2775.2346	14.0815	0.1233	1242.4824	0.0552	1.0000	1	20	108.0870	37879004.3017	17985810.1842	0.4748
276453.0000	2377.1674	12.2868	0.1057	1084.1294	0.0482	10.0000	1	20	108.6910	378790043.0167	47678289.4854	0.1259
288373.0000	2466.6680	12.8166	0.1096	1130.8745	0.0503	100.0000	1	20	108.6911	3787900430.1668	288940000.5234	0.0763
287136.0000	2455.3631	12.7616	0.1091	1126.0235	0.0500	1000.0000	1	20	108.6891	37879004301.6680	2712458033.9583	0.0716
286500.0000	2457.1280	12.7333	0.1092	1123.5294	0.0499	10000.0000	1	20	108.8917	378790043016.6798	26912994542.3390	0.0710

Table A.21: Mandrill, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_\mathbf{u}$	$E_\mathbf{u}/E_{\mathbf{u}^\delta}$
460514.0000	3832.4577	20.4673	0.1703	1805.9373	0.0803	0.0001	1	20	121.1753	73.3701	73.3701	1.0000
460514.0000	3832.4577	20.4673	0.1703	1805.9373	0.0803	0.0010	1	20	121.5570	733.7014	733.7014	1.0000
460517.0000	3832.4638	20.4674	0.1703	1805.9490	0.0803	0.0100	1	20	121.4391	7337.0138	7371.9829	1.0048
460359.0000	3831.7653	20.4604	0.1703	1805.3294	0.0802	0.1000	1	20	121.6327	73370.1380	77739.2918	1.0595
456492.0000	3812.8239	20.2885	0.1695	1790.1647	0.0796	1.0000	1	20	121.2941	733701.3797	811849.6146	1.1065
416851.0000	3591.9514	18.5267	0.1596	1634.7098	0.0727	10.0000	1	20	121.4516	7337013.7965	8093673.5654	1.1031
281829.0000	2443.3978	12.5257	0.1086	1105.2118	0.0491	100.0000	1	20	121.8023	73370137.9650	35512172.9489	0.4840
282936.0000	2437.3141	12.5749	0.1083	1109.5529	0.0493	1000.0000	1	20	122.1400	733701379.6503	196738282.3775	0.2681
287321.0000	2466.1738	12.7698	0.1096	1126.7490	0.0501	10000.0000	1	20	122.2586	7337013796.5034	1817195546.4227	0.2477

Table A.22: Mandrill, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457108.0000	3812.9983	20.3159	0.1695	1792.5804	0.0797	0.0001	1	20	107.7983	3637.6770	3654.6191	1.0047
456261.0000	3808.9173	20.2783	0.1693	1789.2588	0.0795	0.0010	1	20	107.8710	36376.7699	42578.4849	1.1705
389396.0000	3333.4460	17.3065	0.1482	1527.0431	0.0679	0.0100	1	20	107.8605	363767.6990	357034.4768	0.9815
269391.0000	2367.0072	11.9729	0.1052	1056.4353	0.0470	0.1000	1	20	108.2838	3637676.9896	901482.0383	0.2478
263122.0000	2316.0971	11.6943	0.1029	1031.8510	0.0459	1.0000	1	20	108.2200	36376769.8965	4119692.9845	0.1133
266102.0000	2340.2384	11.8268	0.1040	1043.5373	0.0464	10.0000	1	20	108.3807	363767698.9646	35513698.8386	0.0976
265904.0000	2334.2498	11.8180	0.1037	1042.7608	0.0463	100.0000	1	20	108.4643	3637676989.6458	355561734.6782	0.0977
266960.0000	2339.0943	11.8649	0.1040	1046.9020	0.0465	1000.0000	1	20	108.4837	36376769896.4581	3491975194.2898	0.0960
264996.0000	2325.8770	11.7776	0.1034	1039.2000	0.0462	10000.0000	1	20	108.2911	363767698964.5814	35048613519.9341	0.0963

Table A.23: Pirate, Gaussian noise, L^1 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457109.0000	3812.9992	20.3160	0.1695	1792.5843	0.0797	0.0001	1	20	121.7757	71.4703	71.4703	1.0000
457109.0000	3812.9992	20.3160	0.1695	1792.5843	0.0797	0.0010	1	20	120.6558	714.7034	714.7034	1.0000
457097.0000	3812.9381	20.3154	0.1695	1792.5373	0.0797	0.0100	1	20	121.0751	7147.0336	7183.0197	1.0050
455527.0000	3806.0908	20.2456	0.1692	1786.3804	0.0794	0.1000	1	20	121.5545	71470.3361	80755.9473	1.1299
315576.0000	2797.8578	14.0256	0.1243	1237.5529	0.0550	1.0000	1	20	121.4132	714703.3613	645517.9903	0.9032
261448.0000	2301.1415	11.6199	0.1023	1025.2863	0.0456	10.0000	1	20	122.5074	7147033.6132	2456922.9288	0.3438
264218.0000	2321.8316	11.7430	0.1032	1036.1490	0.0461	100.0000	1	20	121.9727	71470336.1316	20350292.4403	0.2847
265351.0000	2328.0775	11.7934	0.1035	1040.5922	0.0462	1000.0000	1	20	121.7716	714703361.3163	196346894.4427	0.2747
264686.0000	2318.5884	11.7638	0.1030	1037.9843	0.0461	10000.0000	1	20	122.5238	7147033613.1626	1991190694.5222	0.2786

Table A.24: Pirate, Gaussian noise, L^1 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457114.0000	3813.0151	20.3162	0.1695	1792.6039	0.0797	0.0001	1	20	107.8616	3637.6770	3652.5823	1.0041
456986.0000	3812.6264	20.3105	0.1695	1792.1020	0.0796	0.0010	1	20	107.5681	36376.7699	39100.7200	1.0749
454533.0000	3800.2575	20.2015	0.1689	1782.4824	0.0792	0.0100	1	20	107.7999	363767.6990	415306.5233	1.1417
427753.0000	3662.9528	19.0112	0.1628	1677.4627	0.0746	0.1000	1	20	107.8892	3637676.9896	4236384.9257	1.1646
307380.0000	2725.8650	13.6613	0.1211	1205.4118	0.0536	1.0000	1	20	107.6312	36376769.8965	17996790.9489	0.4947
259373.0000	2274.4070	11.5277	0.1011	1017.1490	0.0452	10.0000	1	20	108.3105	363767698.9646	54569729.1111	0.1500
266304.0000	2336.3082	11.8357	0.1038	1044.3294	0.0464	100.0000	1	20	108.3625	3637676989.6458	367785009.3960	0.1011
265390.0000	2328.7640	11.7951	0.1035	1040.7451	0.0463	1000.0000	1	20	108.3106	36376769896.4581	3519016250.5893	0.0967
268166.0000	2348.7546	11.9185	0.1044	1051.6314	0.0467	10000.0000	1	20	108.6090	363767698964.5814	35630141344.7630	0.0979

Table A.25: Pirate, Gaussian noise, squared L^2 data term, anisotropic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	α	λ	i	$t[s]$	$E_{\mathbf{u}^\delta}$	$E_{\mathbf{u}}$	$E_{\mathbf{u}}/E_{\mathbf{u}^\delta}$
457109.0000	3812.9992	20.3160	0.1695	1792.5843	0.0797	0.0001	1	20	120.8048	71.4703	71.4703	1.0000
457109.0000	3812.9992	20.3160	0.1695	1792.5843	0.0797	0.0010	1	20	120.9312	714.7034	714.7034	1.0000
457114.0000	3813.0193	20.3162	0.1695	1792.6039	0.0797	0.0100	1	20	120.8651	7147.0336	7183.9754	1.0052
456904.0000	3812.0960	20.3068	0.1694	1791.7804	0.0796	0.1000	1	20	120.9473	71470.3361	75729.4742	1.0596
452541.0000	3791.6597	20.1129	0.1685	1774.6706	0.0789	1.0000	1	20	121.0681	714703.3613	793239.5680	1.1099
407898.0000	3561.3702	18.1288	0.1583	1599.6000	0.0711	10.0000	1	20	121.2301	7147033.6132	7872992.8687	1.1016
261698.0000	2312.5475	11.6310	0.1028	1026.2667	0.0456	100.0000	1	20	121.4443	71470336.1316	35957785.3909	0.5031
263964.0000	2310.5281	11.7317	0.1027	1035.1529	0.0460	1000.0000	1	20	121.9326	714703361.3163	214138596.2316	0.2996
263156.0000	2313.7766	11.6958	0.1028	1031.9843	0.0459	10000.0000	1	20	122.1873	7147033613.1626	1978336833.6229	0.2768

Table A.26: Pirate, Gaussian noise, squared L^2 data term, anisotropic non-quadratic regularization term.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
438638.0000	3692.0000	19.0000	0.0000	1720.0000	0.0000	10.0000	1.0000	2.0000	1373673.0000
438638.0000	3692.0000	19.0000	0.0000	1720.0000	0.0000	5.0000	1.0000	2.0000	1373673.0000
333697.0000	2815.0000	15.0000	0.0000	1309.0000	0.0000	3.0000	1.0000	2.0000	1289123.0000
259725.0000	2243.0000	12.0000	0.0000	1019.0000	0.0000	3.0000	1.0000	4.0000	1153223.0000
209861.0000	2001.0000	9.0000	0.0000	823.0000	0.0000	2.0000	1.0000	3.0000	1016257.0000
184943.0000	1958.0000	8.0000	0.0000	725.0000	0.0000	2.0000	1.0000	6.0000	885520.0000
180411.0000	2024.0000	8.0000	0.0000	707.0000	0.0000	1.0000	1.0000	6.0000	786906.0000
181799.0000	2122.0000	8.0000	0.0000	713.0000	0.0000	1.0000	1.0000	6.0000	712422.0000
187434.0000	2241.0000	8.0000	0.0000	735.0000	0.0000	1.0000	1.0000	5.0000	647432.0000
202639.0000	2443.0000	9.0000	0.0000	795.0000	0.0000	1.0000	1.0000	3.0000	597313.0000
253844.0000	3107.0000	11.0000	0.0000	995.0000	0.0000	1.0000	2.0000	6.0000	646764.0000
371207.0000	4047.0000	16.0000	0.0000	1456.0000	0.0000	1.0000	2.0000	5.0000	789057.0000

Table A.27: Cameraman, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
424827.0000	3595.0000	19.0000	0.0000	1666.0000	0.0000	10.0000	1.0000	2.0000	1336732.0000
424827.0000	3595.0000	19.0000	0.0000	1666.0000	0.0000	5.0000	1.0000	2.0000	1336732.0000
329343.0000	2783.0000	15.0000	0.0000	1292.0000	0.0000	3.0000	1.0000	2.0000	1260365.0000
264390.0000	2280.0000	12.0000	0.0000	1037.0000	0.0000	3.0000	1.0000	4.0000	1137198.0000
228711.0000	2137.0000	10.0000	0.0000	897.0000	0.0000	2.0000	1.0000	3.0000	1013919.0000
220237.0000	2146.0000	10.0000	0.0000	864.0000	0.0000	2.0000	1.0000	8.0000	891063.0000
230945.0000	2307.0000	10.0000	0.0000	906.0000	0.0000	1.0000	1.0000	6.0000	797658.0000
245713.0000	2491.0000	11.0000	0.0000	964.0000	0.0000	1.0000	1.0000	7.0000	726630.0000
263549.0000	2719.0000	12.0000	0.0000	1034.0000	0.0000	1.0000	1.0000	5.0000	663716.0000
285334.0000	2989.0000	13.0000	0.0000	1119.0000	0.0000	1.0000	1.0000	3.0000	614428.0000
343289.0000	3557.0000	15.0000	0.0000	1346.0000	0.0000	1.0000	2.0000	8.0000	672079.0000
462410.0000	4323.0000	21.0000	0.0000	1813.0000	0.0000	1.0000	2.0000	6.0000	860189.0000

Table A.28: Fish, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
458650.0000	3823.0000	20.0000	0.0000	1799.0000	0.0000	10.0000	1.0000	2.0000	1375803.0000
458650.0000	3823.0000	20.0000	0.0000	1799.0000	0.0000	5.0000	1.0000	2.0000	1375803.0000
339795.0000	2832.0000	15.0000	0.0000	1333.0000	0.0000	3.0000	1.0000	3.0000	1285413.0000
255858.0000	2167.0000	11.0000	0.0000	1003.0000	0.0000	3.0000	1.0000	4.0000	1141797.0000
192785.0000	1736.0000	9.0000	0.0000	756.0000	0.0000	2.0000	1.0000	3.0000	999673.0000
157714.0000	1527.0000	7.0000	0.0000	618.0000	0.0000	2.0000	1.0000	6.0000	867239.0000
149818.0000	1544.0000	7.0000	0.0000	588.0000	0.0000	1.0000	1.0000	6.0000	767852.0000
149398.0000	1637.0000	7.0000	0.0000	586.0000	0.0000	1.0000	1.0000	6.0000	693078.0000
152280.0000	1730.0000	7.0000	0.0000	597.0000	0.0000	1.0000	1.0000	7.0000	628257.0000
160864.0000	1922.0000	7.0000	0.0000	631.0000	0.0000	1.0000	1.0000	3.0000	577737.0000
209242.0000	2690.0000	9.0000	0.0000	821.0000	0.0000	1.0000	2.0000	9.0000	617876.0000
323829.0000	3819.0000	14.0000	0.0000	1270.0000	0.0000	1.0000	2.0000	5.0000	710387.0000

Table A.29: House, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
457132.0000	3799.0000	20.0000	0.0000	1793.0000	0.0000	10.0000	1.0000	2.0000	1393336.0000
457132.0000	3799.0000	20.0000	0.0000	1793.0000	0.0000	5.0000	1.0000	2.0000	1393336.0000
345982.0000	2879.0000	15.0000	0.0000	1357.0000	0.0000	3.0000	1.0000	3.0000	1306395.0000
272330.0000	2311.0000	12.0000	0.0000	1068.0000	0.0000	3.0000	1.0000	4.0000	1167777.0000
224782.0000	2026.0000	10.0000	0.0000	881.0000	0.0000	2.0000	1.0000	3.0000	1032332.0000
199664.0000	1866.0000	9.0000	0.0000	783.0000	0.0000	2.0000	1.0000	6.0000	902850.0000
199394.0000	1904.0000	9.0000	0.0000	782.0000	0.0000	1.0000	1.0000	7.0000	806090.0000
205537.0000	1989.0000	9.0000	0.0000	806.0000	0.0000	1.0000	1.0000	7.0000	733365.0000
214510.0000	2082.0000	10.0000	0.0000	841.0000	0.0000	1.0000	1.0000	5.0000	670186.0000
229597.0000	2242.0000	10.0000	0.0000	900.0000	0.0000	1.0000	1.0000	4.0000	621252.0000
287097.0000	2784.0000	13.0000	0.0000	1126.0000	0.0000	1.0000	2.0000	15.0000	687000.0000
513002.0000	4474.0000	23.0000	0.0000	2012.0000	0.0000	1.0000	2.0000	6.0000	869672.0000

Table A.30: Lenna, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
460514.0000	3832.0000	20.0000	0.0000	1806.0000	0.0000	10.0000	1.0000	2.0000	1446728.0000
460514.0000	3832.0000	20.0000	0.0000	1806.0000	0.0000	5.0000	1.0000	2.0000	1446728.0000
353082.0000	2943.0000	16.0000	0.0000	1385.0000	0.0000	3.0000	1.0000	3.0000	1352962.0000
290463.0000	2439.0000	13.0000	0.0000	1139.0000	0.0000	3.0000	1.0000	5.0000	1202400.0000
265582.0000	2279.0000	12.0000	0.0000	1041.0000	0.0000	2.0000	1.0000	3.0000	1053888.0000
254975.0000	2209.0000	11.0000	0.0000	1000.0000	0.0000	2.0000	1.0000	6.0000	912031.0000
260627.0000	2268.0000	12.0000	0.0000	1022.0000	0.0000	1.0000	1.0000	10.0000	805739.0000
268857.0000	2343.0000	12.0000	0.0000	1054.0000	0.0000	1.0000	1.0000	8.0000	725528.0000
278674.0000	2426.0000	12.0000	0.0000	1093.0000	0.0000	1.0000	1.0000	5.0000	656241.0000
289004.0000	2523.0000	13.0000	0.0000	1133.0000	0.0000	1.0000	1.0000	3.0000	602594.0000
329540.0000	2841.0000	15.0000	0.0000	1292.0000	0.0000	1.0000	2.0000	12.0000	641524.0000
461279.0000	3793.0000	21.0000	0.0000	1809.0000	0.0000	1.0000	2.0000	5.0000	739399.0000

Table A.31: Mandrill, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
457109.0000	3813.0000	20.0000	0.0000	1793.0000	0.0000	10.0000	1.0000	2.0000	1413924.0000
457109.0000	3813.0000	20.0000	0.0000	1793.0000	0.0000	5.0000	1.0000	2.0000	1413924.0000
345828.0000	2883.0000	15.0000	0.0000	1356.0000	0.0000	3.0000	1.0000	3.0000	1324561.0000
274083.0000	2322.0000	12.0000	0.0000	1075.0000	0.0000	3.0000	1.0000	4.0000	1182261.0000
240799.0000	2119.0000	11.0000	0.0000	944.0000	0.0000	2.0000	1.0000	3.0000	1044645.0000
221486.0000	1994.0000	10.0000	0.0000	869.0000	0.0000	2.0000	1.0000	8.0000	912344.0000
226965.0000	2061.0000	10.0000	0.0000	890.0000	0.0000	1.0000	1.0000	8.0000	813256.0000
236173.0000	2148.0000	10.0000	0.0000	926.0000	0.0000	1.0000	1.0000	8.0000	738790.0000
249439.0000	2261.0000	11.0000	0.0000	978.0000	0.0000	1.0000	1.0000	5.0000	673974.0000
265668.0000	2406.0000	12.0000	0.0000	1042.0000	0.0000	1.0000	1.0000	4.0000	623614.0000
325983.0000	2916.0000	14.0000	0.0000	1278.0000	0.0000	1.0000	2.0000	11.0000	683979.0000
517564.0000	4335.0000	23.0000	0.0000	2030.0000	0.0000	1.0000	2.0000	6.0000	844397.0000

Table A.32: Pirate, Gaussian noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
287242.0000	6708.0000	13.0000	0.0000	1126.0000	0.0000	10.0000	1.0000	2.0000	1409614.0000
287242.0000	6708.0000	13.0000	0.0000	1126.0000	0.0000	5.0000	1.0000	2.0000	1409614.0000
89301.0000	3210.0000	4.0000	0.0000	350.0000	0.0000	3.0000	1.0000	2.0000	1264441.0000
47896.0000	1353.0000	2.0000	0.0000	188.0000	0.0000	3.0000	1.0000	4.0000	1056246.0000
73093.0000	1442.0000	3.0000	0.0000	287.0000	0.0000	2.0000	1.0000	3.0000	905795.0000
87099.0000	1541.0000	4.0000	0.0000	342.0000	0.0000	2.0000	1.0000	4.0000	790186.0000
99335.0000	1696.0000	4.0000	0.0000	390.0000	0.0000	1.0000	1.0000	5.0000	702464.0000
110982.0000	1875.0000	5.0000	0.0000	435.0000	0.0000	1.0000	1.0000	4.0000	639202.0000
124260.0000	2041.0000	6.0000	0.0000	487.0000	0.0000	1.0000	1.0000	4.0000	584636.0000
139631.0000	2284.0000	6.0000	0.0000	548.0000	0.0000	1.0000	1.0000	3.0000	542586.0000
195445.0000	3038.0000	9.0000	0.0000	766.0000	0.0000	1.0000	2.0000	13.0000	601741.0000
287777.0000	3980.0000	13.0000	0.0000	1129.0000	0.0000	1.0000	2.0000	5.0000	767471.0000

Table A.33: Cameraman, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
284560.0000	7072.0000	13.0000	0.0000	1116.0000	0.0000	10.0000	1.0000	2.0000	1455929.0000
284560.0000	7072.0000	13.0000	0.0000	1116.0000	0.0000	5.0000	1.0000	2.0000	1455929.0000
93076.0000	3471.0000	4.0000	0.0000	365.0000	0.0000	3.0000	1.0000	2.0000	1316359.0000
54486.0000	1538.0000	2.0000	0.0000	214.0000	0.0000	3.0000	1.0000	4.0000	1117938.0000
76755.0000	1535.0000	3.0000	0.0000	301.0000	0.0000	2.0000	1.0000	3.0000	975405.0000
94488.0000	1681.0000	4.0000	0.0000	371.0000	0.0000	2.0000	1.0000	4.0000	863199.0000
112470.0000	1924.0000	5.0000	0.0000	441.0000	0.0000	1.0000	1.0000	6.0000	778073.0000
134281.0000	2245.0000	6.0000	0.0000	527.0000	0.0000	1.0000	1.0000	4.0000	714516.0000
155908.0000	2475.0000	7.0000	0.0000	611.0000	0.0000	1.0000	1.0000	5.0000	657611.0000
181805.0000	2761.0000	8.0000	0.0000	713.0000	0.0000	1.0000	1.0000	3.0000	613584.0000
252574.0000	3399.0000	11.0000	0.0000	990.0000	0.0000	1.0000	2.0000	11.0000	689456.0000
375754.0000	4180.0000	17.0000	0.0000	1474.0000	0.0000	1.0000	2.0000	6.0000	918164.0000

Table A.34: Fish, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
292726.0000	6468.0000	13.0000	0.0000	1148.0000	0.0000	10.0000	1.0000	2.0000	1329039.0000
292726.0000	6468.0000	13.0000	0.0000	1148.0000	0.0000	5.0000	1.0000	2.0000	1329039.0000
86559.0000	3140.0000	4.0000	0.0000	339.0000	0.0000	3.0000	1.0000	2.0000	1183784.0000
33046.0000	994.0000	1.0000	0.0000	130.0000	0.0000	3.0000	1.0000	4.0000	975909.0000
45315.0000	927.0000	2.0000	0.0000	178.0000	0.0000	2.0000	1.0000	2.0000	828157.0000
51730.0000	910.0000	2.0000	0.0000	203.0000	0.0000	2.0000	1.0000	4.0000	720734.0000
60137.0000	1007.0000	3.0000	0.0000	236.0000	0.0000	1.0000	1.0000	4.0000	639278.0000
69326.0000	1146.0000	3.0000	0.0000	272.0000	0.0000	1.0000	1.0000	4.0000	581473.0000
81066.0000	1319.0000	4.0000	0.0000	318.0000	0.0000	1.0000	1.0000	4.0000	531218.0000
97630.0000	1627.0000	4.0000	0.0000	383.0000	0.0000	1.0000	1.0000	3.0000	492927.0000
152756.0000	2444.0000	7.0000	0.0000	599.0000	0.0000	1.0000	2.0000	15.0000	542249.0000
250624.0000	3593.0000	11.0000	0.0000	983.0000	0.0000	1.0000	2.0000	4.0000	659840.0000

Table A.35: House, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
293580.0000	6559.0000	13.0000	0.0000	1151.0000	0.0000	10.0000	1.0000	2.0000	1433900.0000
293580.0000	6559.0000	13.0000	0.0000	1151.0000	0.0000	5.0000	1.0000	2.0000	1433900.0000
92093.0000	3151.0000	4.0000	0.0000	361.0000	0.0000	3.0000	1.0000	2.0000	1287246.0000
48706.0000	1316.0000	2.0000	0.0000	191.0000	0.0000	3.0000	1.0000	4.0000	1077667.0000
67767.0000	1240.0000	3.0000	0.0000	266.0000	0.0000	2.0000	1.0000	3.0000	929317.0000
76927.0000	1251.0000	3.0000	0.0000	302.0000	0.0000	2.0000	1.0000	5.0000	816212.0000
87665.0000	1352.0000	4.0000	0.0000	344.0000	0.0000	1.0000	1.0000	6.0000	731864.0000
97115.0000	1456.0000	4.0000	0.0000	381.0000	0.0000	1.0000	1.0000	4.0000	670504.0000
109042.0000	1566.0000	5.0000	0.0000	428.0000	0.0000	1.0000	1.0000	4.0000	617888.0000
124328.0000	1741.0000	6.0000	0.0000	488.0000	0.0000	1.0000	1.0000	3.0000	577428.0000
180655.0000	2326.0000	8.0000	0.0000	708.0000	0.0000	1.0000	2.0000	13.0000	661184.0000
402048.0000	4061.0000	18.0000	0.0000	1577.0000	0.0000	1.0000	2.0000	6.0000	906050.0000

Table A.36: Lenna, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
291689.0000	6340.0000	13.0000	0.0000	1144.0000	0.0000	10.0000	1.0000	2.0000	1576471.0000
291689.0000	6340.0000	13.0000	0.0000	1144.0000	0.0000	5.0000	1.0000	2.0000	1576471.0000
114398.0000	3092.0000	5.0000	0.0000	449.0000	0.0000	3.0000	1.0000	2.0000	1424636.0000
92084.0000	1390.0000	4.0000	0.0000	361.0000	0.0000	3.0000	1.0000	5.0000	1202659.0000
138572.0000	1632.0000	6.0000	0.0000	543.0000	0.0000	2.0000	1.0000	3.0000	1032678.0000
162036.0000	1740.0000	7.0000	0.0000	635.0000	0.0000	2.0000	1.0000	6.0000	894515.0000
178995.0000	1865.0000	8.0000	0.0000	702.0000	0.0000	1.0000	1.0000	4.0000	791903.0000
192812.0000	1966.0000	9.0000	0.0000	756.0000	0.0000	1.0000	1.0000	4.0000	715757.0000
207467.0000	2073.0000	9.0000	0.0000	814.0000	0.0000	1.0000	1.0000	5.0000	650149.0000
223893.0000	2212.0000	10.0000	0.0000	878.0000	0.0000	1.0000	1.0000	3.0000	599620.0000
267038.0000	2535.0000	12.0000	0.0000	1047.0000	0.0000	1.0000	2.0000	11.0000	649081.0000
393491.0000	3437.0000	17.0000	0.0000	1543.0000	0.0000	1.0000	2.0000	5.0000	785264.0000

Table A.37: Mandrill, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
278161.0000	6367.0000	12.0000	0.0000	1091.0000	0.0000	10.0000	1.0000	2.0000	1433009.0000
278161.0000	6367.0000	12.0000	0.0000	1091.0000	0.0000	5.0000	1.0000	2.0000	1433009.0000
95629.0000	3095.0000	4.0000	0.0000	375.0000	0.0000	3.0000	1.0000	2.0000	1295484.0000
57293.0000	1233.0000	3.0000	0.0000	225.0000	0.0000	3.0000	1.0000	4.0000	1096901.0000
85734.0000	1295.0000	4.0000	0.0000	336.0000	0.0000	2.0000	1.0000	3.0000	952019.0000
102661.0000	1380.0000	5.0000	0.0000	403.0000	0.0000	2.0000	1.0000	4.0000	836423.0000
115800.0000	1507.0000	5.0000	0.0000	454.0000	0.0000	1.0000	1.0000	6.0000	750251.0000
128713.0000	1625.0000	6.0000	0.0000	505.0000	0.0000	1.0000	1.0000	4.0000	687159.0000
143506.0000	1758.0000	6.0000	0.0000	563.0000	0.0000	1.0000	1.0000	5.0000	631926.0000
161138.0000	1944.0000	7.0000	0.0000	632.0000	0.0000	1.0000	1.0000	3.0000	589878.0000
220137.0000	2478.0000	10.0000	0.0000	863.0000	0.0000	1.0000	2.0000	11.0000	668882.0000
421886.0000	4020.0000	19.0000	0.0000	1654.0000	0.0000	1.0000	2.0000	6.0000	890697.0000

Table A.38: Pirate, Salt & Pepper noise, L^1 data term, anisotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
329697.0000	2791.0000	15.0000	0.0000	1293.0000	0.0000	10.0000	1.0000	14.0000	51393145.0000
242323.0000	2141.0000	11.0000	0.0000	950.0000	0.0000	2.0000	1.0000	28.0000	26283879.0000
236455.0000	2264.0000	11.0000	0.0000	927.0000	0.0000	1.0000	1.0000	43.0000	17234043.0000
243205.0000	2447.0000	11.0000	0.0000	954.0000	0.0000	1.0000	2.0000	54.0000	19677459.0000
287699.0000	3077.0000	13.0000	0.0000	1128.0000	0.0000	1.0000	2.0000	103.0000	25668251.0000

Table A.39: Cameraman, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
321057.0000	2732.0000	14.0000	0.0000	1259.0000	0.0000	10.0000	1.0000	14.0000	50106003.0000
244225.0000	2177.0000	11.0000	0.0000	958.0000	0.0000	2.0000	1.0000	31.0000	26282195.0000
246430.0000	2369.0000	11.0000	0.0000	966.0000	0.0000	1.0000	1.0000	49.0000	17426178.0000
260908.0000	2589.0000	12.0000	0.0000	1023.0000	0.0000	1.0000	2.0000	58.0000	19989319.0000
325146.0000	3246.0000	14.0000	0.0000	1275.0000	0.0000	1.0000	2.0000	101.0000	26242503.0000

Table A.40: Fish, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
340048.0000	2840.0000	15.0000	0.0000	1334.0000	0.0000	10.0000	1.0000	17.0000	47580611.0000
221478.0000	1873.0000	10.0000	0.0000	869.0000	0.0000	2.0000	1.0000	29.0000	23711631.0000
199685.0000	1792.0000	9.0000	0.0000	783.0000	0.0000	1.0000	1.0000	46.0000	15314539.0000
198567.0000	1885.0000	9.0000	0.0000	779.0000	0.0000	1.0000	2.0000	57.0000	17316605.0000
226203.0000	2372.0000	10.0000	0.0000	887.0000	0.0000	1.0000	2.0000	103.0000	21987519.0000

Table A.41: House, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
341703.0000	2843.0000	15.0000	0.0000	1340.0000	0.0000	10.0000	1.0000	15.0000	49258163.0000
230504.0000	1974.0000	10.0000	0.0000	904.0000	0.0000	2.0000	1.0000	29.0000	24902826.0000
216034.0000	1955.0000	10.0000	0.0000	847.0000	0.0000	1.0000	1.0000	50.0000	16289728.0000
221526.0000	2078.0000	10.0000	0.0000	869.0000	0.0000	1.0000	2.0000	60.0000	18612446.0000
274899.0000	2636.0000	12.0000	0.0000	1078.0000	0.0000	1.0000	2.0000	105.0000	24521557.0000

Table A.42: Lenna, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
345503.0000	2875.0000	15.0000	0.0000	1355.0000	0.0000	10.0000	1.0000	15.0000	51480261.0000
244574.0000	2054.0000	11.0000	0.0000	959.0000	0.0000	2.0000	1.0000	29.0000	24895494.0000
235783.0000	2013.0000	10.0000	0.0000	925.0000	0.0000	1.0000	1.0000	46.0000	15705360.0000
241765.0000	2083.0000	11.0000	0.0000	948.0000	0.0000	1.0000	2.0000	59.0000	17473021.0000
276750.0000	2406.0000	12.0000	0.0000	1085.0000	0.0000	1.0000	2.0000	96.0000	21312178.0000

Table A.43: Mandrill, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_{\mathbf{u}}$
341670.0000	2854.0000	15.0000	0.0000	1340.0000	0.0000	10.0000	1.0000	14.0000	50396547.0000
236287.0000	2006.0000	11.0000	0.0000	927.0000	0.0000	2.0000	1.0000	26.0000	25296466.0000
228301.0000	1999.0000	10.0000	0.0000	895.0000	0.0000	1.0000	1.0000	40.0000	16426571.0000
237351.0000	2119.0000	11.0000	0.0000	931.0000	0.0000	1.0000	2.0000	59.0000	18659910.0000
293164.0000	2643.0000	13.0000	0.0000	1150.0000	0.0000	1.0000	2.0000	110.0000	24155567.0000

Table A.44: Pirate, Gaussian noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
290079.0000	5035.0000	13.0000	0.0000	1138.0000	0.0000	10.0000	1.0000	17.0000	132464573.0000
290724.0000	3331.0000	13.0000	0.0000	1140.0000	0.0000	2.0000	1.0000	29.0000	62815041.0000
297030.0000	3021.0000	13.0000	0.0000	1165.0000	0.0000	1.0000	1.0000	45.0000	39169520.0000
304762.0000	3018.0000	14.0000	0.0000	1195.0000	0.0000	1.0000	2.0000	57.0000	43239572.0000
339186.0000	3394.0000	15.0000	0.0000	1330.0000	0.0000	1.0000	2.0000	110.0000	51625954.0000

Table A.45: Cameraman, Salt & Pepper noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
292566.0000	5341.0000	13.0000	0.0000	1147.0000	0.0000	10.0000	1.0000	14.0000	143924835.0000
306637.0000	3580.0000	14.0000	0.0000	1202.0000	0.0000	2.0000	1.0000	29.0000	68721850.0000
323248.0000	3269.0000	14.0000	0.0000	1268.0000	0.0000	1.0000	1.0000	45.0000	42968926.0000
338477.0000	3272.0000	15.0000	0.0000	1327.0000	0.0000	1.0000	2.0000	60.0000	47476421.0000
393278.0000	3638.0000	17.0000	0.0000	1542.0000	0.0000	1.0000	2.0000	104.0000	56702390.0000

Table A.46: Fish, Salt & Pepper noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
282864.0000	4820.0000	13.0000	0.0000	1109.0000	0.0000	10.0000	1.0000	15.0000	118178478.0000
257921.0000	2993.0000	11.0000	0.0000	1011.0000	0.0000	2.0000	1.0000	31.0000	55590524.0000
249552.0000	2536.0000	11.0000	0.0000	979.0000	0.0000	1.0000	1.0000	48.0000	34471531.0000
249490.0000	2445.0000	11.0000	0.0000	978.0000	0.0000	1.0000	2.0000	69.0000	37906664.0000
269074.0000	2649.0000	12.0000	0.0000	1055.0000	0.0000	1.0000	2.0000	106.0000	44733810.0000

Table A.47: House, Salt & Pepper noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
290243.0000	4907.0000	13.0000	0.0000	1138.0000	0.0000	10.0000	1.0000	15.0000	122981811.0000
278042.0000	3140.0000	12.0000	0.0000	1090.0000	0.0000	2.0000	1.0000	29.0000	58070563.0000
280286.0000	2753.0000	12.0000	0.0000	1099.0000	0.0000	1.0000	1.0000	46.0000	36146516.0000
288361.0000	2709.0000	13.0000	0.0000	1131.0000	0.0000	1.0000	2.0000	55.0000	39882849.0000
336553.0000	3037.0000	15.0000	0.0000	1320.0000	0.0000	1.0000	2.0000	106.0000	47706489.0000

Table A.48: Lenna, Salt & Pepper noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
299627.0000	4750.0000	13.0000	0.0000	1175.0000	0.0000	10.0000	1.0000	20.0000	116537563.0000
290286.0000	3059.0000	13.0000	0.0000	1138.0000	0.0000	2.0000	1.0000	30.0000	54370815.0000
286106.0000	2656.0000	13.0000	0.0000	1122.0000	0.0000	1.0000	1.0000	47.0000	33447336.0000
287908.0000	2564.0000	13.0000	0.0000	1129.0000	0.0000	1.0000	2.0000	55.0000	36562755.0000
308682.0000	2658.0000	14.0000	0.0000	1211.0000	0.0000	1.0000	2.0000	99.0000	42440157.0000

Table A.49: Mandrill, Salt & Pepper noise, squared L^2 data term, isotropic TV.

$\|\mathbf{u}^0-\mathbf{u}\|_1$	$\|\mathbf{u}^0-\mathbf{u}\|_2$	$\|\mathbf{u}^0-\mathbf{u}\|_1/len(\mathbf{u}^0)$	$\|\mathbf{u}^0-\mathbf{u}\|_2/len(\mathbf{u}^0)$	r	$\bar{r}$	γ	α	$t[s]$	$E_\mathbf{u}$
282095.0000	4774.0000	13.0000	0.0000	1106.0000	0.0000	10.0000	1.0000	15.0000	116881131.0000
278739.0000	3082.0000	12.0000	0.0000	1093.0000	0.0000	2.0000	1.0000	29.0000	55312620.0000
285110.0000	2720.0000	13.0000	0.0000	1118.0000	0.0000	1.0000	1.0000	48.0000	34459433.0000
294816.0000	2678.0000	13.0000	0.0000	1156.0000	0.0000	1.0000	2.0000	61.0000	38032987.0000
341778.0000	2978.0000	15.0000	0.0000	1340.0000	0.0000	1.0000	2.0000	97.0000	45481007.0000

Table A.50: Pirate, Salt & Pepper noise, squared L^2 data term, isotropic TV.