HIGHER ORDER LEVELABLE MRF ENERGY MINIMIZATION VIA GRAPH CUTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET HAYDAR KARCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2008

Approval of the thesis

"HIGHER ORDER LEVELABLE MRF ENERGY MINIMIZATION VIA GRAPH CUTS"

submitted by **Mehmet Haydar Karcı** in partial fullfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering** by,

Prof. Dr. Canan Özgen ______________________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ______________________
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Mübeccel Demirekler ______________________
Supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu ______________________
Electrical and Electronics Engineering, METU

Prof. Dr. Mübeccel Demirekler ______________________
Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Yasemin Yardımcı Çetin ______________________
Informatics Institute, METU

Assoc. Prof. Dr. Gözde Bozdağı Akar ______________________
Electrical and Electronics Engineering, METU

Prof. Dr. Enis Çetin ______________________
Electrical and Electronics Engineering, Bilkent University

Date: February 28, 2008

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mehmet Haydar Karcı

Signature :

ABSTRACT

HIGHER ORDER LEVELABLE MRF ENERGY MINIMIZATION VIA GRAPH CUTS

Karcı, Mehmet Haydar
Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mübeccel Demirekler

February 2008, 80 pages

A feature of minimizing images of a class of binary Markov random field energies is introduced and proved. Using this, the collection of minimizing images of levels of higher order, levelable MRF energies is shown to be a monotone collection. This implies that these images can be combined to give minimizing images of the MRF energy itself. Due to the recent developments, second and third order binary MRF energies of the mentioned class are known to be exactly minimized by maximum flow/minimum cut computations on appropriately constructed graphs. With the aid of these developments an exact and efficient algorithm to minimize levelable second and third order MRF energies, which is composed of a series of maximum flow/minimum cut computations, is proposed and applications of the proposed algorithm to image restoration are given.

Keywords: Markov Random Fields, Image Restoration, Network Flows, Graph Cuts, Maximum Flow, Minimum Cut, Levelable Energies

ÖZ

ÇİZGE KESİLERİ TEMELLİ YÜKSEK DERECELİ DÜZEYLENEBİLİR MRA MİNİMİZASYONU

Karcı, Mehmet Haydar
Doktora, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mübeccel Demirekler

ŞUBAT 2008, 80 sayfa

İkili Markov rasgele alan enerjilerinin bir türünü küçülten görüntülerin bir özelliği sunuldu ve ispatlandı. Bu özellik kullanılarak yüksek dereceli, düzeylenebilir MRA enerjilerinin düzeylerini küçülten görüntüler yığınının monoton bir yığın olduğu ve bu görüntülerin MRA enerjisini küçültmek üzere birleştirilebileceği gösterildi. Bahsi geçen ikili MRA enerjilerinin ikinci ve üçüncü dereceden olanlarının uygun tasarlanmış çizgeler üzerinde uygulanan maksimum akış/minimum kesi algoritmalarıyla kesin ve etkili biçimde küçültülebileceği son gelişmelerle bilinmektedir. Bunların yardımıyla ikinci ve üçüncü dereceden düzeylenebilir MRA enerjilerini küçülten ve bir dizi maksimum akış/minimum kesi hesaplamasından oluşan bir algoritma önerildi ve bu algoritmanın görüntü iyileştirme üzerine bazı uygulamaları sunuldu.

Anahtar Kelimeler: Markov Rasgele Alanları, Görüntü İyileştirme, Ağ Akışları, Çizge Kesileri, Maksimum Akış, Minimum Kesi, Düzeylenebilir Enerjiler

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Many image processing problems are essentially estimation (inversion) problems. Denoising problem viewed as estimation of the original image before contamination by noise or moving object tracking problem viewed as estimation of the motion vectors are examples. Estimation on the other hand, always goes in hand with optimization. In statistical inversion [20] for example, estimation is almost immediately recast into maximization of a probability density function. In addition, many estimation problems are presented as optimization problems at the very beginning. Least-squares methods, as examples of deterministic approaches to inversion, are commonly formulated as optimization problems themselves. This approach relies on a cost (energy) function which encodes our information about the estimatee and minimization of it to gather the best solution. We refer to the text books [9, 2] for examples of optimization or energy minimization based methods in various fields of image processing.

One important issue in energy minimization framework is the construction of the energy function. Energy function should summarize our information about the estimated quantity and, since we deal with minimizing it, should be a measure of how far a candidate is from the desired solution. Traditionally energy functions are presented in two parts, data fidelity and prior. The data fidelity part forces the solution to keep close to the observed data and the prior, which typically does not depend on the data, forces it to obey the restrictions which summarize our prior information about it. Apart from the expressiveness of them, energy functions should be constructed regarding computational issues as well. Existence and uniqueness of minimizers or ease of minimization are also important aspects of the problem. Minimization of the energy function, despite the fact that there is a whole optimization theory literature to the assistance, is also an effortful part of the energy minimization framework. One of the problems here is the scarcity of global minimization algorithms. Most optimization algorithms are impaired with the possibility of getting stuck with local minima. Another problem is the efficiency which becomes very crucial especially with large scale and time critical problems which are typical in image processing.

In the third chapter, we are going to introduce an algorithm which minimizes the following energy function

$$F(x) = \sum_{i \in \Sigma} f_i(x_i, y_i) + \sum_{i,j \in \Sigma} f_{ij}(x_i, x_j) + \sum_{i,j,k \in \Sigma} f_{ijk}(x_i, x_j, x_k) \tag{1.1}$$

for certain types of functions f_i, f_{ij} and f_{ijk}[1]. Here Σ denotes the lattice of pixels, x and y denote the estimated and observed images, and the intensities associated to ith pixel of the images x and y are denoted by x_i and y_i respectively. Minimization of the energy function above can be used in various fields of image processing like restoration, inpainting etc. In Section 3.2.5, when we present numerical examples of our method, we are going to deal with denoising problems.

We encounter energy functions like the one given in (1.1) in the theory of Markov Random Fields, MRFs hereinafter, where the maximum a posteriori estimation is equivalent to an energy minimization. Hence as we are going to review in the following chapter, MRFs provide the Bayesian justification of the energy minimization framework, so we prefer to refer to the energy function (1.1) as MRF energy. Besides that, MRF formulation is not essential for the rest of the material in this thesis.

We assume that the intensities of the noisy and the estimated images are integers. Due to the operation of image sensing devices, it is common practice to assume integer valued intensities for noisy images. Assuming integer values

[1] Our intention is to give a rather informal introduction in this chapter. We leave the details of the notation to Chapter 2 and Section 3.2.1.

for the intensities of the estimated images is also encouraged by the image displaying devices, however it is not as common. Usually the intensities of the estimated images are assumed to be real numbers and they are rounded to integers for displaying. In the sequel we assume that the estimated images are also composed of integer intensities. This places the minimization problem in a highly complicated and, in some aspects, harder class of problems, namely integer programming. However quite surprisingly, recent devolopments lead to efficient minimization schemes of a class of MRF energies of this kind with the assistance from a seemingly unrelated area, network flow theory. The methods which depend on this collaboration are said to be *graph-cuts* methods. We are going to give an overview of graph-cuts based devolopments in image processing in the next chapter.

Graph-cuts methods can be roughly categorized into two classes, exact and approximate. As we already pointed out that graph-cuts methods apply for only a class of MRF energies and one can correctly guess that exact methods require more restrictions on the energy functions compared to approximate ones. Some of the exact methods require smaller graphs than the others to provide more efficient algorithms. Needless to say these more efficient algorithms apply for a highly restricted class of MRF energies.

In this thesis we study exact and efficient minimization of MRF energies using graph-cut techniques. Second chapter is an introduction to MRFs, where we introduce the related notation and definitions. We also introduce some new (up to our knowledge) structures in this chapter. In the third chapter we propose an efficient algorithm which exactly minimizes MRF energies with convex data fidelity terms and *levelable* prior terms. In the introductory section of the third chapter we start with an overview of the graph-cuts literature and we summarize our contribution to the subject. The following section, Section 3.2 is devoted to binary MRFs. We first present an overview of the literature where network flow theory and minimization of binary MRF energies meet. Then we start our treatment of binary MRFs with introducing a new notation and a set of definitions in Section 3.2.1. With the aid of these, we present Theorem 3.15, a new feature of minimizers of higher order, regular binary MRF energies in the following Section 3.2.2. In this section we also give a notation which substantially simplifies dealing with higher order functions. We start Section 3.2.3 introducing a partial ordering of functions of binary variables. This is going to help us to give a generalization of a known property, namely monotonicity, of minimizers of binary MRF energies.

In Section 3.2.4 we make use of the theory developed in Section 3.2. Here we define levelable functions and introduce our algorithm which minimizes levelable, higher order MRF energies. Finally we provide applications of our algorithm to image denoising in Section 3.2.5. We are going to conclude and discuss possible future directions in Chapter 4.

To summarize, this thesis contributes to the subject in terms of

- introducing a useful notation and providing new, comprehensive definitions,
- presenting a new feature of minimizers of binary MRF energies, which we think may be of further use even on its own,
- introducing an abstraction to the theory of binary MRFs and generalization of monotonicity property of minimizers of binary MRF energies to higher order energies,
- introducing an algorithm which efficiently minimizes higher order, levelable MRF energies and
- providing applications of higher order, levelable priors to image restoration.

CHAPTER 2

AN INTRODUCTION TO MRF MODELS

We justify our energy minimization approach by estimation of MRFs. Since in MRF modelling maximum a posteriori estimation is equivalent to an energy minimization [25], once we model image signals as instances MRFs, energy minimization becomes our natural choice for estimation. Besides, another class of methods, namely variational methods in image processing [9], can easily be cast into the MRF framework [20] after discretization. Hence the MRF framework is a fairly general one for image processing and has attracted great attention in almost all fields of image processing. For extensive treatments of MRFs see [7, 24, 32] and see [27] for a review. The text books [9, 20] also have sections on MRFs.

In this chapter we give a very brief introduction to MRFs following the texts [7, 24] to which we refer for the details we skip. For the following assume that L is a positive, finite integer.

Definition 2.1 *We define the set $\Lambda_L = \{0,..,L-1\}$, which we call the **intensity space**. Let Σ be a non-empty, finite set of elements called **pixels**. The family $\Lambda_L^\Sigma = \{x : \Sigma \to \Lambda_L\}$ is called the **configuration space** and each $x \in \Lambda_L^\Sigma$ is called an **image**. Each item in the intensity space Λ_L is said to be a **intensity**.*

The terminology given above is not standard. Usually the terms site, phase space and configuration are used instead of pixel, intensity space and image respectively.

Notation 2.2 *For any nonnegative integer n, we use Λ_L^n to denote the set of*

n-tuples of intensities. Note that $\Lambda_L^0 = \varnothing$.

Notation 2.3 *We set $\Sigma = \{1, .., M\}$, hence $\Lambda_L^\Sigma = \Lambda_L^M$ with no loss of generality. Therefore, each image $x \in \Lambda_L^\Sigma$ is denoted by the M-tuple $(x_1, .., x_M)$ where $x_s = x(s)$ for $s = 1, .., M$. We denote $x(S) = (x_s)_{s \in S}$ for a given subset S of Σ.*

Definition 2.4 *A **neighborhood system** on Σ is a family $\mathcal{N} = \{\mathcal{N}_s\}_{s \in \Sigma}$ of subsets of Σ so that*

$$\begin{aligned} i. &\quad s \notin \mathcal{N}_s \\ ii. &\quad s \in \mathcal{N}_t \Longleftrightarrow t \in \mathcal{N}_s \end{aligned}$$

*for any $s, t \in \Sigma$. The set $\mathcal{N}_s$ is said to be the **neighborhood** of the pixel s. We denote $\widetilde{\mathcal{N}}_s = \mathcal{N}_s \cup s$. The couple $(\Sigma, \mathcal{N})$ is called a **topology**.*

We define the following topology related structures for future reference. The last two are new as far as we know.

Definition 2.5 *Given a topology $(\Sigma, \mathcal{N})$, the **boundary** ∂S of $S \subset \Sigma$ is defined as*

$$\partial S = (\bigcup_{s \in S} \mathcal{N}_s) \setminus S$$

Definition 2.6 *Given a topology $(\Sigma, \mathcal{N})$, any two pixels s and t in $S \subset \Sigma$ are said to be **connected** in S, if there exists a collection of pixels $\{s_i\}_{i=0}^r$ in S, where $s = s_0$ and $t = s_r$ such that*

$$s_{i+1} \in \widetilde{\mathcal{N}}_{s_i}$$

*for $0 \leq i < r - 1$. A set $S \subset \Sigma$ is called **connected** if any two pixels in S are connected in S.*

Definition 2.7 *Given a topology $(\Sigma, \mathcal{N})$ and an image $x \in \Lambda_L^\Sigma$. Any connected set $C \in \Sigma$ is said to be a **component** with respect to x if $x_s = x_t$ for any $s, t \in C$. Then we say $x_C = x_s$, for any $s \in C$, is **well defined**.*

In the sequel, any image in the configuration space is going to be assumed an instance of a random field, which we define next.

Definition 2.8 *A **random field** X on Σ with intensities in Λ_L is a collection $X = (X_1, .., X_M)$ of random variables which take values in Λ_L.*

According to the definition above, for each pixel $s \in \Sigma$, x_s is an instance of the random variable X_s. However, a random field can also be viewed as a random variable taking values in the configuration space. With this interpretation each instance of a random field is an image.

Notation 2.9 *We define $X(S) = (X_s)_{s \in S}$ for a given subset S of Σ.*

Definition 2.10 *A random field on Σ is called a **Markov random field** (MRF) with respect to the neighborhood system $\mathcal{N}$ if*

i. $P(X = x) > 0$

ii. $P\left(X_s = x_s \mid X(\Sigma \setminus s) = x(\Sigma \setminus s)\right) = P\left(X_s = x_s \mid X(\mathcal{N}_s) = x(\mathcal{N}_s)\right)$

for all $s \in \Sigma$ and $x \in \Lambda_L^\Sigma$.

The first equation above is called the **positivity condition** [24] and is a technical requirement. Among the various versions, to avoid unnecessary details and since it causes no essential loss of generality for our purposes, we stick to this one. The second equation makes MRFs valuable for signal processing. According to it, statistics of any pixel may depend on any other pixel however this dependence can only be through its neighbor pixels. Futhermore this compromise between accuracy and simplification can be fine tuned by the neighborhood structure.

Definition 2.11 *Given a topology $(\Sigma, \mathcal{N})$, a set $\pi \subset \Sigma$ is called **clique** if either π is a singleton or any pixel in π is a neighbor of any other pixel in π.*

We discriminate cliques with the number of elements they have. An n-clique or a clique of order n refers to a clique of n pixels. The collection of cliques of Σ is determined when a neigborhood system on Σ is given. Conversely if a family of cliques of Σ is given, a feasible neighborhood system can be identified. Indeed, for the trivial neighborhood system, where any pixel is a neighbor of any other pixel, all subsets of Σ are cliques.

Notation 2.12 *We denote the family of all n-cliques on Σ by $\Pi^n(\Sigma)$ or shortly, by Π^n.*

Definition 2.13 *A **Gibbs potential** on Λ_L^Σ relative to the neighborhood system*

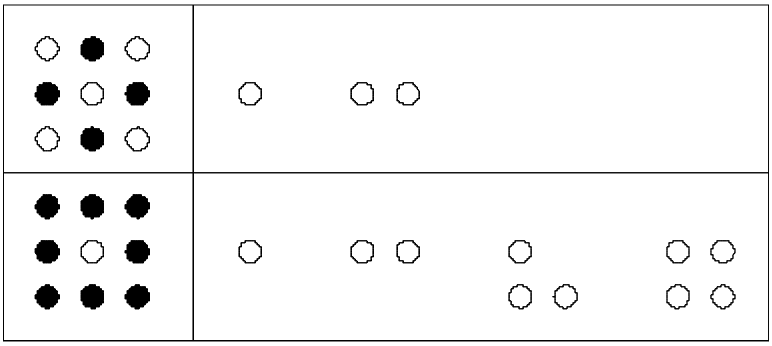

Figure 2.1: Two commonly used topologies. The circles represent pixels. Black circles are neighbors of the circle in the middle. The topologies in the first column are called 4-neighborhood topology and 8-neighborhood topology respectively. The associated cliques, up to rotation, are given on the right.

$\mathcal{N}$ is a collection $\{g_\pi\}_{\pi\subset\Sigma}$, of functions $g_\pi : \Lambda_L^\Sigma \to \mathbb{R}$, where

$$i. \quad g_\pi \equiv 0, \quad \textit{if } \pi \textit{ is not a clique} \tag{2.1}$$

$$ii. \quad x(\pi) = y(\pi) \Longrightarrow g_\pi(x) = g_\pi(y) \tag{2.2}$$

for all x,y in Λ_L^Σ and all $\pi \subset \Sigma$.

The following theorem is known as Hammersley-Clifford theorem. Its proof may be found in the texts [7, 32].

Theorem 2.14 *A random field is an MRF with respect to the neighborhood system $\mathcal{N}$ if and only if its joint density function $p(x) : \Lambda_L^\Sigma \to (0, 1)$ is given by*

$$p(x) = \frac{1}{Z} exp(-\frac{1}{T} F(x)) \tag{2.3}$$

where T, Z are constants and $F : \Lambda_L^\Sigma \to \mathbb{R}$ is a function given by

$$F(x) = \sum_{\pi\subset\Sigma} g_\pi(x) \tag{2.4}$$

where the collection $\{g_\pi\}_{\pi\subset\Sigma}$ is a Gibbs potential relative to $\mathcal{N}$.

Remarks

1. The constant T is called **temperature**. Other constant Z is a normalizing constant and is called **partition function**. We call $F(x)$ the **MRF energy** of x and each g_π, $\pi \subset \Sigma$, a **potential function**.

2. Let K denote the maximum number of pixels a clique can have, given the topology. Then because of (2.1) we can rewrite (2.4) as

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} g_\pi(x)$$

Thence Equation (2.2) implies that

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k}) \tag{2.5}$$

where, for any clique $\pi \subset \Pi^n$, $x(\pi)$ is denoted by $(x_{\pi_1}, .., x_{\pi_n})$ and the function $f_\pi : \Lambda_L^n \to \mathbb{R}$ is defined as

$$f_\pi(x_{\pi_1}, .., x_{\pi_n}) = g_\pi(x)$$

For MRF energies we prefer to use the form given by (2.5) and for any $\pi \in \Pi^n$ we call each f_π an n-th order potential function. We also refer to MRF energies with the order of the largest clique, thus for instance, the one given in (2.5) is a K-th order MRF energy. □

The following example demonstrates how MRFs are used in a typical image processing problem, denoising.

Example 2.15 Let $z = (z_1, .., z_M)$, where $z_s \in \Lambda_L$ for $1 \leq s \leq M$, be a given image. Assume that z is a degraded and then quantized version of an image $x = (x_1, .., x_M)$ which we want to estimate. Say, we presume the 4-neighborhood topology (see Figure 2.1) for x and pose the following energy function.

$$F(x) = \sum_{\pi \in \Pi^1} f_\pi(x_{\pi_1}) + \sum_{\pi \in \Pi^2} f_\pi(x_{\pi_1}, x_{\pi_2})$$

As F is an MRF energy, the joint density of x is given by Equation (2.3). Hence a maximum a posteriori estimate of x is a maximizer of (2.3) or equivalently, a minimizer of F.

This observation gives a hint about how to choose the functions f_π. Usually if $\pi \in \Pi^1$, f_π is chosen to be a high-pass function which penalizes deviations between x_{π_1} and z_{π_1}. On the other hand the second term of F does not depend on the noisy observation and somehow reflects the characteristics of the image signal up to our information or inclination. The scope of this thesis does not cover the issue of how to embed the prior information or inclination in the energy function. For additional information about this topic one can see [20].

Usually, the part of F which depends on measured data is called the **data fidelity** energy and the part of F which depends on the estimated signal only is called the **prior** energy. Although of course, there is no theoretical obligation to separate these terms. $\square$

Denoising is one of the typical estimation problems in image processing. Any estimation problem, which can be put as a minimization of an energy function like the one we have given in (2.5), therefore, is an MRF problem hence falls in the scope of this thesis. However, in the sequel we are going to have to limit ourselves to certain data fidelity and prior energies.

Notation 2.16 *We define* $V_L^n = \{f : \Lambda_L^n \rightarrow \mathbb{R} \mid f(0,..,0) = 0\}$ *for positive* n*.*

Note that as we are concerned with minimizing (2.5), assuming $f(0,..,0) = 0$ causes no loss of generality. Any function f of n variables in (2.5) can safely be replaced with $\hat{f}$ defined as

$$\hat{f}(x_1,..,x_n) = f(x_1,..,x_n) - f(0,..,0)$$

In fact for the rest of this text, the assumption $f(0,..,0) = 0$ is an unessential detail.

CHAPTER 3

MINIMIZATION OF MRF ENERGIES

In the previous chapter we justified our energy minimization approach with the aid of MRF modelling of image signals. In this chapter we deal with the minimization of MRF energies. In the following section we present a survey of graph-cuts based MRF energy minimization literature and we give a summary of our contribution in Section 3.1.1. The following section, Section 3.2 is the core of this work. In this section we start with a review of the equivalence of the binary MRF energy minimization and maximum flow problems. Then we introduce an additional notation in 3.2.1 and introduce a new feature of minimizers of binary MRF energies in 3.2.2. Next we deal with the monotonicity property of minimizers in 3.2.3 and in 3.2.4 we extend the theory to general MRF energies. We finally give some numerical examples in image denoising in Section 3.2.5.

3.1 Introduction

One class of MRF energy minimization problems is of special importance, binary MRF energy minimization. Minimization of a class of this type of energy functions is known to be equivalent to the maximum flow problem [1, 3] in an appropriately defined graph since 1970's [28, 17]. The maximum flow formulation of such problems lead to exact and efficient algorithms for energy minimization for binary MRFs, in contrast to the other existing algorithms which lacked either of these properties. See [17] for a comparison of a maximum flow based approach with two traditional algorithms, simulated annealing [14] and ICM [4] in binary image restoration[1]. As the scope of these algorithms was underestimated to cover only binary image processing, their utilization was limited.

The results of [28, 17] are recently extended by Kolmogorov et al. in [22] where the class of binary MRF energies which can be represented by an appropriate graph is called *graph representable* and necessary and sufficient conditions for graph representability of second and third order MRF energy functions are given. Later, their results are reestablished with simpler algebraic arguments in [13]. In [22] the graph constructions for the second and third order graph representable MRF energies are also given. The case with higher order energies is addressed in both of these papers and also in [26, 34]. In the sequel we deal with a class of MRF energies of any finite order which can be decomposed into binary MRF energies of the same order. For the minimization of the resulting binary MRF energies we just refer to [22]. Thus for the applications we are limited up to third order energies for the time being. However our algorithm does not depend on the order of the energy, so the adaptation of it for energies of order higher than three is trivial.

Today, efficent MRF energy minimization algorithms exist and a variety of them is based on maximum flow formulation. The revival of maximum flow based algorithms for MRF energies is due to the recognition of the fact that MRF energies could be rewritten, decomposed or recast in terms of maximum flow problems. MRF energy minimization methods depending on this recognition are called, by convention, graph-cut methods [6]. See [31] for a comparison of MRF energy minimization methods, including graph-cut based methods.

The first class[2] of graph-cut implementations are due to [29] and [19] for stereo and image restoration respectively. These implementations base on rewriting a class of second order MRF energies as maximum flow problems[3] on appropriately defined graphs. That representation however, requires a huge graph, typically

[1] See [14] for simulated annealing, [4] for ICM or [27] for a review of both.

[2] This simple classification is not chronological and is meant to be according to the similarities of the implementations.

[3] In [29] there is no explicit reference to MRFs though.

one node for each possible intensity level for each pixel in [19] for example, so its implementation is not as effective. Similar treatments can also be found in [11, 33].

The second class of implementations [10, 18, 8, 33, 16] addresses the huge graph issue. There, a class of MRF energies is decomposed into several binary MRF energies. Then using the minimizers of these binary MRFs, a minimizer of the initial MRF energy is constructed. This approach required several maximum flow implementations on much smaller graphs; for a comparison, typically one node for each pixel, to build more effective implementations of highly limited number of MRF energies.

The first two classes of graph-cut methods share an important property that they provide exact minimizers of MRF energies. This is not the case for the third class of methods presented in [6] where Boykov et al. presented two descent like iterative minimization schemes. In each iteration they recast the problem of finding the maximum amount of descent in the energy as a maximum flow problem. The ensuing algorithms are not exact, however in [6] they also show that exact minimization problem of even some of the most basic MRF energies is NP hard, hence virtually impossible. The class of energies for which these methods could be applied is much wider than those of the other two classes of methods. The size of the required graph and the number of times a maximum flow calculation is performed depend on the case.

We base our approach on the paper [10] of Darbon and Sigelle, hence the second class of methods. Recall that similar algorithms also developed in [18, 8, 33, 16]. In order to clarify our contribution to the subject, we would like to give a little detail of their work.

Recall that $\Sigma = \{1, .., M\}$ is the configuration space, i.e., a finite lattice of pixels. Remember that L denotes the number of intensity levels, which is typically 256, and $\Lambda_L = \{0, .., L-1\}$ denotes the intensity space. Then the MRF energy function is given by (2.5)

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

where Π^k denotes the family of cliques of order k. For any $\pi = \{\pi_1, .., \pi_k\} \in \Pi^k$, f_π is the associated potential function. The variable $x = (x_1, .., x_M) \in \Lambda_L^\Sigma$ denotes an image. The integer K denotes the order of the largest cliques. The aim is to find a minimizer of F.

In [10], Darbon and Sigelle developed an algorithm to minimize a variant of F. They took $K = 2$ and they assumed

i. f_π is a convex function for which $f(0) = 0$ holds[4], if $\pi \in \Pi^1$

ii. $f_\pi(x_{\pi_1}, x_{\pi_2}) = a_{ij}|x_{\pi_1} - x_{\pi_2}|$ if $\pi \in \Pi^2$ and $a_{ij} \geq 0$

The idea behind the algorithm is to decompose F into levels as follows

$$\begin{aligned} F(x) &= \sum_{i=0}^{L-2} F^i(x^i) \\ &= \sum_{i=0}^{L-2} \Big(\sum_{\pi \in \Pi^1} [f_\pi(i+1) - f_\pi(i)] x^i_{\pi_1} + \sum_{\pi \in \Pi^2} a_{ij} |x^i_{\pi_1} - x^i_{\pi_2}| \Big) \end{aligned}$$

where for any nonnegative integer $y < L$

$$y^i = \begin{cases} 1 & : \quad i < y \\ 0 & : \quad i \geq y \end{cases}$$

for $i = 0, .., L-2$[5]. Note that for each i, F^i is a binary MRF in terms of the *level sets* x^i of the image x. They proved that there exists minimizers of F^i for $i = 0, .., L-2$, which are the level sets of a minimizer of F. They calculated each minimizer of F^i using a maximum flow computation on an appropriately designed graph as proposed in [22] to construct a minimizer of F. They also gave a binary search type implementation which required $log(L)$ maximum flow computations.

The possibility of such a decomposition is due to the levelability of the absolute value of differences of integer pairs into levels, i.e., for nonnegative integers

[4] The $f(0) = 0$ requirement is appended by us just for the sake of the clarity of the notation.
[5] This notation is a slightly altered version of the one given in [10].

$y, z < L$

$$|y - z| = \sum_{i=0}^{L-2} |y^i - z^i|$$

Indeed, the method given by [10] applies to any second order potential f_π as long as it is levelable, i.e.,

$$\begin{aligned} &i. \quad f_\pi(x_{\pi_1}, x_{\pi_2}) = \sum_{i=0}^{L-2} f_\pi^i(x_{\pi_1}^i, x_{\pi_2}^i) \\ &ii. \quad f_\pi^i = f_\pi^j \text{ for } 0 \leq i, j \leq L-2 \end{aligned} \tag{3.1}$$

The definition of levelability is in fact due to a later paper of the same authors [11]. Moreover in their definition they did not assume (3.1). However the implementation they gave for the levelable functions for which (3.1) does not hold, is not the one they gave in their first paper [10] and is not as efficient.

3.1.1 Our Contribution

In this work we follow a generic approach. We start with a property of minimizers of regular binary MRFs. Using it we reach a generalization of the method given in [10] in two aspects.

i. We generalize the method to higher order MRFs.

ii. We relax the notion of levelability, namely the Equation (3.1), such that the same method still applies for minimization of the MRF energy.

To our knowledge the issue of higher order potentials has never been addressed for this class of graph-cut based methods. For the other classes of graph cut based methods, the issue of higher order energies is addressed in [34, 21]. We think, quite reasonably that, using higher order priors better representation of image signals may be acquired. As we previously mentioned, the relaxed levelability has also been considered by Darbon and Sigelle in [11], but the implementation they gave falls into the first class of graph-cut based methods rather than the second.

3.2 Binary Markov Random Fields

In the previous section we introduced MRFs and how they are utilized in estimation problems in image processing. In Section 3.2.4 we are going to present an algorithm which efficiently minimizes a class of MRF energy functions. This algorithm is going to be based some properties of minimizers of binary MRFs hence first we have to deal with them.

We first present an overview of minimization of binary MRFs via maximum flow algorithms. Our aim is only to provide an intuition how these two seemingly unrelated problems are equivalent. We refer to the text books [1, 3] for detailed treatments of maximum flow/minimum cut problems and to the papers [22, 13] for rigorous establishment of equivalence of binary MRF minimization and maximum flow problems.

The main objects of interest in network flow theory are directed graphs. A **(directed) graph** $G(V, A)$ consists of

- a finite set of **nodes**, V,
- a set of ordered pairs of nodes, A.

We assume that there are two distinguished **terminal nodes** in the graph, denoted by s and t. We denote $V = \{s, t, 1, .., M\}$, where M is a positive integer denoting the number of non-terminal nodes. Each item $(u, v) \in A$ is said to be an **arc** and is associated with a nonnegative **capacity** $c(u, v)$[6]. A **flow** on $G(V, A)$ is a mapping $p : A \to \mathbb{Z}$ for which the following conditions hold

$$
\begin{aligned}
&i. \quad 0 \leq p(u, v) \leq c(u, v), \quad \text{for any } (u, v) \in A \\
&ii. \quad \sum_{(u,v)\in A} p(u, v) - \sum_{(v,u)\in A} p(v, u) = 0, \quad \text{for any } u \in V \setminus \{s, t\}
\end{aligned}
$$

The quantity $\sum_{(s,u)\in A} p(s, u)$ is said to be the **value** of the flow. A pair of disjoint sets of nodes (S, T) for which $S \cup T = V$ holds, is said to be a **cut** if $s \in S$ and $t \in T$. The capacity of the cut $c(S, T)$ is defined as

$$
c(S, T) = \sum_{\substack{(u,v)\in A \\ u\in S, v\in T}} c(u, v)
$$

[6] We assume integer capacities with no essential loss of generality.

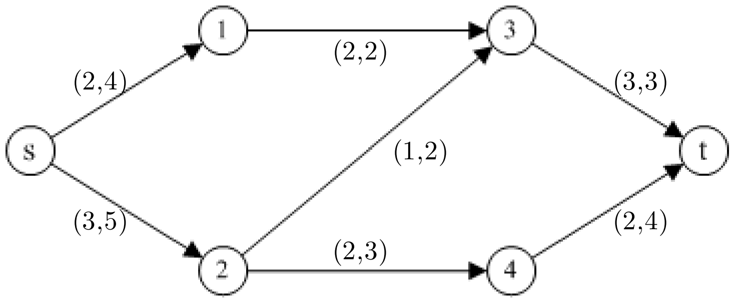

Figure 3.1: A simple graph and a flow defined on it. The pairs of integers accompanying the arcs are the associated flows $p(u, v)$ and the capacities $c(u, v)$ in this order.

Determination of the maximum possible value of the flow on a given graph is the celebrated maximum flow problem in integer programming. There exist efficient, polynomial time algorithms for maximum flow problems. One important thing to note is that, those algorithms also solve another important problem in integer programming, namely the minimum cut problem which deals with identifying the cut with the minimum capacity on a given graph. Minimum cut problem constitutes the connection between the maximum flow problem and the minimization of binary MRF energies as follows[7].

Let $x \in \Lambda_2^{\Sigma}$ and consider the following second order binary MRF energy function.

$$F(x) = \sum_{\pi \in \Pi^1} f_\pi(x_{\pi_1}) + \sum_{\pi \in \Pi^2} f_\pi(x_{\pi_1}, x_{\pi_2}) \tag{3.2}$$

Note that we can write

$$\begin{aligned} f_\pi(x_{\pi_1}, x_{\pi_2}) = f_\pi(0,0)(1 - x_{\pi_1})(1 - x_{\pi_2}) + f_\pi(0,1)(1 - x_{\pi_1})x_{\pi_2} \\ + f_\pi(1,0)x_{\pi_1}(1 - x_{\pi_2}) + f_\pi(1,1)x_{\pi_1}x_{\pi_2} \end{aligned}$$

for any $\pi \in \Pi^2$ and

$$f_\pi(x_{\pi_1}) = f_\pi(0)(1 - x_{\pi_1}) + f_\pi(1)x_{\pi_1}$$

[7] We adapt the following argument from [13] to which we refer for the full treatment.

for any $\pi \in \Pi^1$. Thence

$$F(x) = \sum_{\pi \in \Pi^2} [f_\pi(0,0) + f_\pi(1,1) - f_\pi(0,1) - f_\pi(1,0)] x_{\pi_1} x_{\pi_2} + L$$
$$= \sum_{\pi \in \Pi^2} [f_\pi(0,1) + f_\pi(1,0) - f_\pi(0,0) - f_\pi(1,1)] x_{\pi_1} (1 - x_{\pi_2}) + L'$$

where L and L' are affine functions of x. Let us rewrite F as follows

$$F(x) = \sum_{\pi \in \Pi^2} a_\pi x_{\pi_1}(1 - x_{\pi_2}) + \sum_{i \in \Sigma} a_i x_i + c \tag{3.3}$$

where for any $\pi \in \Pi^2$

$$a_\pi = f_\pi(0,1) + f_\pi(1,0) - f_\pi(0,0) - f_\pi(1,1)$$

a_i is appropriately defined for any $i \in \Sigma$[8] and c is a constant. Assume that $a_\pi \geq 0$ for any $\pi \in \Pi^2$.

Now let us construct a graph with one node for each pixel in Σ in addition to two terminal nodes and enumerate non-terminal nodes with the indices of the corresponding pixels. Match any given cut to an image according to the following rule.

- For any node $u \in S$, set the intensity of the corresponding pixel to 0
- For any node $v \in T$, set the intensity of the corresponding pixel to 1

This way any cut is associated with a unique image and vice versa. For any $\pi \in \Pi^2$ append an arc (π_2, π_1) to the graph with capacity a_π[9]. For any $i \in \Sigma$ append an arc (s, i) with capacity a_i if $a_i > 0$ and append an arc (i, t) with capacity $-a_i$ if $a_i \leq 0$. Let (S, T) be any cut in this graph. Notice that due to the definition of a cut, for any $\pi \in \Pi^2$, the contribution of the arc (π_2, π_1) to the capacity of the cut is nonzero only if $x_{\pi_2} = 0$ and $x_{\pi_1} = 1$. Therefore for any $\pi \in \Pi^2$, the contribution of the arc (π_2, π_1) to the capacity of the cut is $a_\pi x_{\pi_1}(1 - x_{\pi_2})$. On the other hand the contribution of the arc (s, i) to the capacity of the cut is $a_i x_i$ if $a_i > 0$ and zero otherwise[10]. Similarly the contribution of the

[8] For the following we do not need the explicit form of a_i.
[9] This is why we require $a_\pi \geq 0$.
[10] We maintain here that an arc with zero capacity is equivalent to a non existant arc.

arc (i,t) to the capacity of the cut is $-a_i(1-x_i)$ if $a_i \leq 0$ and zero otherwise. To sum up we have

$$c(S,T) = \sum_{\pi\in\Pi^2} a_\pi x_{\pi_1}(1-x_{\pi_2}) + \sum_{i\in\Sigma:a_i>0} a_i x_i + \sum_{i\in\Sigma:a_i\leq 0} -a_i(1-x_i)$$

which is equal to the MRF energy given in (3.3) up to a constant. This verifies that minimization of the binary MRF energy function given by (3.3) or by (3.2) is equivalent to the minimum cut problem on the graph we defined. In turn, this problem is already known to be equivalent to the maximum flow problem on the same graph. In the sequel we are going to decompose a given non-binary MRF energy into binary MRF energies. When we need to minimize each binary MRF energy to minimize the given non-binary energy we started with, we are going to use this equivalence.

In the following section we shall present our notation and give some definitons. In the next section our aim is to explore a characteristic of the minimizers of binary MRF energies. The trailing section is devoted to a very important property of minimizers of binary MRFs, monotonicity. Thanks to this property we are going to be able to decompose a class of MRF energies in terms of binary MRF energies and construct the aforesaid algorithm in the last section.

3.2.1 Notation and Definitions

Although the concepts we describe are simple, the notation we have to use can get assorted and hard to follow. As a remedy for this, after the definitions or results which may seem complicated, we present examples which hopefully reduce the complication.

Throughout this section let n be a positive integer and for any $y \in \Lambda_2$, let y' denote the negation of y, i.e., $y' = 1 - y$. We also denote $I_n = \{1, .., n\}$.

Definition 3.1 *Define integers p, q, m so that $0 \leq p \leq n$, $p + q = n$ and $0 \leq m \leq q$. Let $\{q_1, .., q_m\}$ be a set of nonnegative integers such that $\sum_{i=1}^{m} q_i = q$. Define the ordered collection of integers $\alpha^0 = \{\alpha^0_1, .., \alpha^0_p\}$ and define the sets of integers $\alpha^i = \{\alpha^i_1, .., \alpha^i_{q_i}\}$ with $i = 1, .., m$, so that the collection $\{\alpha^0, .., \alpha^m\}$*

disjointly splits I_n*, i.e.*

$$\alpha^i \cap \alpha^j = \varnothing, \quad 0 \leq i \neq j \leq m$$

$$\bigcup_{i=0}^{m} \alpha^i = I_n$$

Then we call the ordered collection $\{\alpha^0; \alpha^1, .., \alpha^m\}$*, a* $(p; q_1, .., q_m)$ ***partition*** *of* I_n*.*

Although we used the set notation for α^0, we have to emphasize that α^0 is an ordered collection of integers from I_n, i.e., we distinguish between same sets with different orderings for α^0. This makes no difference for this definition but is important for the following. For the sets α^i with $i > 0$ ordering is not important. Similarly, note that the collection of sets $\{\alpha^0; \alpha^1, .., \alpha^m\}$ is also defined to be an ordered collection.

Example 3.2 The ordered collection $\{\alpha^0; \alpha^1, \alpha^2\}$, where $\alpha^0 = \{4\}$, $\alpha^1 = \{1, 2\}$ and $\alpha^2 = \{3\}$, is a $(1; 2, 1)$ partition of I_4. Note that the word *ordered* is important, as for example neither $\{\alpha^0; \alpha^2, \alpha^1\}$ nor $\{\alpha^1; \alpha^0, \alpha^2\}$ is a $(1; 2, 1)$ partition of I_4. □

Example 3.3 The ordered collection $\{\alpha^0; \alpha^1, \alpha^2\}$, where $\alpha^0 = \varnothing$, $\alpha^1 = \{1, 2\}$ and $\alpha^2 = \{3\}$, is a $(0; 2, 1)$ partition of I_3. □

The following definition is an extended version of the one given in [22]. It simply formalizes the notion of projection of a function of n variables into a function of m variables with $m \leq n$. We split the arguments of the function into $m + 1$ groups. We keep the items in the first group constant and constrain the items in the other groups mutually equal. The latter constraint is going to be of use when we define regularities of functions of binary variables.

Definition 3.4 *Let the collection* $\{\alpha^0; \alpha^1, .., \alpha^m\}$ *be a* $(p; q_1, .., q_m)$ *partition of* I_n*. Given* $a = (a_1, .., a_p) \in \Lambda_L^p$*, we define the operator*

$$\mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1,..,\alpha^m} : V_L^n \to V_L^n$$

such that

$$\mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1,..,\alpha^m}(f)(\hat{y}_1, .., \hat{y}_m) = f(y_1, .., y_n)$$

where

$$y_{\alpha^0_k} = a_k, \quad k = 1, .., p$$

$$y_{\alpha^j_k} = \hat{y}_j, \quad j = 1, .., m, \quad k = 1, .., q_j$$

We say $\mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1,..,\alpha^m}(f)$ *is the* $(p; q_1, .., q_m)$ ***projection*** *of* f *in* m *degrees of freedom onto sets* $\alpha^0, .., \alpha^m$ *with respect to* $(a_1, .., a_p)$.

If $p = 0$, we denote the $(0; q_1, .., q_m)$ projection of f with $\mathcal{P}^{\varnothing}_{\varnothing;\alpha^1,..,\alpha^m}(f)$.

Example 3.5 Let $f \in V_2^5$. The function $\mathcal{P}^{(a_1 a_2)}_{\alpha^0;\alpha^1,\alpha^2}(f)$ with

$$\mathcal{P}^{(a_1 a_2)}_{\alpha^0;\alpha^1,\alpha^2}(f)(\hat{y}_1, \hat{y}_2) = f(\hat{y}_1, a_1, \hat{y}_1, \hat{y}_2, a_2)$$

is the $(2; 2, 1)$ projection of f in two degrees of freedom onto sets $\alpha^0 = \{2, 5\}$, $\alpha^1 = \{1, 3\}$, $\alpha^2 = \{4\}$ with respect to $a = (a_1, a_2)$. Note that if $\beta^0 = \{5, 2\}$, i.e., the same set as α^0 with altered ordering, then

$$\mathcal{P}^{(a_1 a_2)}_{\beta^0;\alpha^1,\alpha^2}(f)(\hat{y}_1, \hat{y}_2) \neq \mathcal{P}^{(a_1 a_2)}_{\alpha^0;\alpha^1,\alpha^2}(f)(\hat{y}_1, \hat{y}_2)$$

in general. □

Example 3.6 Let $f \in V_L^3$. The function $\mathcal{P}^{\varnothing}_{\varnothing;\alpha^1,\alpha^2}(f)$ with

$$\mathcal{P}^{\varnothing}_{\varnothing;\alpha^1,\alpha^2}(f)(\hat{y}_1, \hat{y}_2) = f(\hat{y}_1, \hat{y}_2, \hat{y}_1)$$

is the $(0; 2, 1)$ projection of f onto sets $\alpha^0 = \varnothing$, $\alpha^1 = \{1, 3\}$ and $\alpha^2 = \{2\}$. □

Next, we define the regularities of a function of binary variables. In a way, a regularity is a spatial measure of how high-pass the function in consideration is.

Definition 3.7 *Let the collection* $\{\alpha^0; \alpha^1, \alpha^2\}$ *be a* $(p; q_1, q_2)$ *partition of* I_n*, let* $a = (a_1, .., a_p) \in \Lambda_2^p$ *and* $f \in V_2^n$*. We define the operator*

$$\mathcal{R}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2} : V_2^n \to \mathbb{R}$$

as

$$\mathcal{R}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2}(f) = \hat{f}(0, 1) + \hat{f}(1, 0) - \hat{f}(0, 0) - \hat{f}(1, 1)$$

where $\hat{f} = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2}(f)$*. The quantity* $\mathcal{R}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2}(f)$ *is said to be the* $(p; q_1, q_2)$ ***regularity*** *of* f *onto sets* $\alpha^0, \alpha^1, \alpha^2$ *with respect to* $(a_1, .., a_p)$.

If $p = 0$ we use the notation $\mathcal{R}^{\varnothing}_{\varnothing;\alpha^1,\alpha^2}(f)$. Note that if either of the sets α^1 or α^2 is empty, $\mathcal{R}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2} = 0$. Thus all regularities of functions of single variable are necessarily zero.

Example 3.8 The quantity

$$\mathcal{R}^{(a_1)}_{\alpha^0;\alpha^1,\alpha^2}(f) = f(0,0,1,a_1) + f(1,1,0,a_1) - f(0,0,0,a_1) - f(1,1,1,a_1)$$

is the $(1;2,1)$ regularity of $f \in V_2^4$ onto sets $\alpha^0 = \{4\}$, $\alpha^1 = \{1,2\}$, $\alpha^2 = \{3\}$ with respect to $a = (a_1)$. □

Functions which have nonnegative regularities is of greater importance.

Definition 3.9 *A function $f \in V_2^n$ is said to be **regular** if all regularities of f are nonnegative.*

Remarks

1. Functions of single variable are, by definition, regular.

2. This definition of regularity of functions of binary variables is equivalent to the one given in [22]. We prove this in Appendix A. □

3.2.2 A Property of Minimizers

Definition 3.10 *For a given set $S \subset \Sigma$ we define the **inversion operator** $\kappa_S : \Lambda_2^\Sigma \to \Lambda_2^\Sigma$ as follows*

$$z = \kappa_S(x) \triangleq \kappa_S \cdot x \Longleftrightarrow \begin{cases} z_i{=}x_i & : \quad i \notin S \\ z_i{=}x_i' & : \quad i \in S \end{cases}$$

We denote composition $\kappa_S \circ \kappa_T$ of two operators κ_S, κ_T by κ_{ST}.

Recall the MRF energy

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

For the following analysis, we need to split the above energy to smaller pieces. First we define

$$F^k(x) = \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

Therefore the cliques involved in $F^k(x)$ are k-cliques only.

Notation 3.11 *Let m be a positive integer. For any given collection of disjoint subsets $S_1, .., S_m$ of Σ and a collection of nonnegative integers $u_1, .., u_m$ such that $k = u_1 + .. + u_m$, we define $\Pi^{ku_1..u_m}_{S_1,..,S_m}$ to be the set of all k-cliques exactly u_j pixels of which belong to S_j with $j = 1, .., m$.*

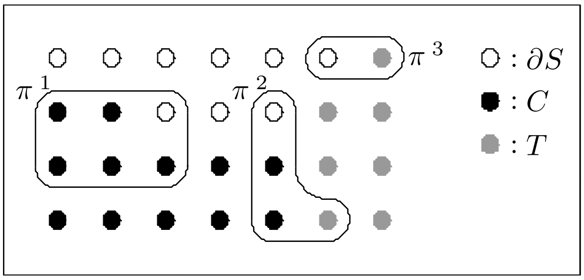

Figure 3.2: An illustration of Notation 3.11 for example 3.12

Example 3.12 See Figure 3.2, where we assumed $S = C \cup T$ and $C \cap T = \varnothing$. Therefore, $\pi^1 \in \Pi^{651}_{C,\partial S}$, $\pi^2 \in \Pi^{4211}_{C,T,\partial S}$, $\pi^3 \in \Pi^{211}_{T,\partial S}$. □

The following notation helps us to simplify the notation a bit further.

Notation 3.13 *We define*

$$e^{ku_1..u_m}_{S_1,..,S_m} = \sum_{\pi \in \Pi^{ku_1..u_m}_{S_1,..,S_m}} f_\pi(x(\pi))$$

and we introduce the following convention. If either of the sets, S_i for instance, is primed in the above notation, this would mean that the image x underwent the inversion κ_{S_i} before the function evaluation. For example

$$e^{ku_1..u_i..u_m}_{S_1,..,S'_i,..,S_m} = \sum_{\pi \in \Pi^{ku_1..u_m}_{S_1,..,S_m}} f_\pi((\kappa_{S_i} \cdot x)(\pi))$$

From now on, assume that the summation indices types of which are not clear in the context, are nonnegative integers. Let S be a subset of Σ, define

$$F^k_S(x) = \sum_{\substack{u+v=k \\ u \neq 0}} e^{kuv}_{S,\partial S}$$

and

$$F^k_{\bar{S}}(x) = F^k(x) - F^k_S(x)$$

Notice that $F^k_{\bar{S}}(x)$ is comprised of all and only the terms of $F^k(x)$ which are not related to the pixels in S [11]. Define

$$F^k_{S'}(x) = F^k(\kappa_S \cdot x) - F^k_{\bar{S}}(x)$$

hence

$$F^k_{S'}(x) = \sum_{\substack{u+v=k \\ u\neq 0}} e^{kuv}_{S',\partial S}$$

Therefore we have

$$F_S(x) \triangleq \sum_{k=1}^{K} F^k_S(x) = \sum_{k=1}^{K} \sum_{\substack{u+v=k \\ u\neq 0}} e^{kuv}_{S,\partial S}$$

and

$$F_{S'}(x) \triangleq \sum_{k=1}^{K} F^k_{S'}(x) = \sum_{k=1}^{K} \sum_{\substack{u+v=k \\ u\neq 0}} e^{kuv}_{S',\partial S}$$

Note that for any pair of images x, y, we have

$$\begin{aligned} F(y) - F(x) &= F(\kappa_S \cdot x) - F(x) \\ &= F_{S'}(x) - F_S(x) \end{aligned}$$

since there exists some subset S of Σ for which $y = \kappa_S \cdot x$.

Definition 3.14 *We define* ***gain due to*** *S, $\Delta F_S(x)$, as the amount of energy increase caused by inverting the intensities of the sites in $S \subset \Sigma$, i.e.*

$$\Delta F_S(x) = F_{S'}(x) - F_S(x)$$

In the sequel, we are going to investigate the term $\Delta F_S(x)$ in finer details. To this end we shall decompose the gain $\Delta F_S(x)$ in terms of $\Delta F_C(x)$ and $\Delta F_T(x)$, where the sets C and T disjointly split S. First we deal with $F^k_S(x)$

$$\begin{aligned} F^k_S(x) &= \sum_{\substack{u+v=k \\ u\neq 0}} e^{kuv}_{S,\partial S} \\ &= \sum_{\substack{u+v=k \\ u\neq 0}} \sum_{u'+v'=u} e^{ku'v'v}_{C,T,\partial S} \end{aligned} \tag{3.4}$$

[11] Note also that $F^k_{\Sigma\setminus S}(x) \neq F^k_{\bar{S}}(x)$ in general.

Notice that for $k = u + v$

$$e^{kuv}_{C,\partial C} = \sum_{v'=0}^{v} e^{kuv'(k-u-v')}_{C,T,\partial S}$$

therefore

$$e^{ku0v}_{C,T,\partial S} = e^{kuv}_{C,\partial C} - \sum_{v'=1}^{v} e^{kuv'(k-u-v')}_{C,T,\partial S}$$

and similarly

$$e^{ku0v}_{T,C,\partial S} = e^{kuv}_{T,\partial T} - \sum_{v'=1}^{v} e^{kuv'(k-u-v')}_{T,C,\partial S}$$

Note that $e^{ku'v'v}_{C,T,\partial S} = e^{kv'u'v}_{T,C,\partial S}$ as long as $u = u' + v'$, thus rewriting (3.4) we have

$$\begin{aligned}
F^k_S(x) &= \sum_{\substack{u+v=k\\u\neq 0}} \{e^{ku0v}_{C,T,\partial S} + e^{k0uv}_{C,T,\partial S} + \sum_{\substack{u'+v'=u\\u'\neq 0\\v'\neq 0}} e^{ku'v'v}_{C,T,\partial S}\} \\
&= \sum_{\substack{u+v=k\\u\neq 0}} \{e^{kuv}_{C,\partial C} + e^{kuv}_{T,\partial T}\} + I^k_{S,C,T}(x) \\
&= F^k_C(x) + F^k_T(x) + I^k_{S,C,T}(x) \qquad (3.5)
\end{aligned}$$

where

$$\begin{aligned}
I^k_{S,C,T}(x) &= \sum_{\substack{u+v=k\\u\neq 0}} \{ \sum_{\substack{u'+v'=u\\u'\neq 0\\v'\neq 0}} e^{ku'v'v}_{C,T,\partial S} - \sum_{v'=1}^{v} \left(e^{kuv'(k-u-v')}_{C,T,\partial S} + e^{kuv'(k-u-v')}_{T,C,\partial S}\right)\} \\
&= \sum_{\substack{u+v=k\\u\neq 0}} \sum_{\substack{u'+v'=u\\u'\neq 0\\v'\neq 0}} e^{ku'v'v}_{C,T,\partial S} - \sum_{\substack{u+v=k\\u\neq 0}} \sum_{v'=1}^{v} \left(e^{kuv'(k-u-v')}_{C,T,\partial S} + e^{kuv'(k-u-v')}_{T,C,\partial S}\right) \\
&= \sum_{\substack{u+v=k\\u>1}} \sum_{\substack{u'+v'=u\\u'\neq 0\\v'\neq 0}} e^{ku'v'v}_{C,T,\partial S} - \sum_{u=1}^{k-1} \sum_{v'=1}^{k-u} \left(e^{kuv'(k-u-v')}_{C,T,\partial S} + e^{kuv'(k-u-v')}_{T,C,\partial S}\right)
\end{aligned}$$

For the first term in the equation above note that, if $u = 1 = u' + v'$, either one of u', v' has to be zero. For the second term in the last equation, note that for $u = k$, the corresponding summation in the preceding equation does not exist. Simplifying further we have

$$I^k_{S,C,T}(x) = \sum_{\substack{u'+v'=2\\u'\neq 0\\v'\neq 0}}^{k} e^{ku'v'v}_{C,T,\partial S} - \sum_{\substack{u+v'=2\\u\neq 0\\v'\neq 0}}^{k} \left(e^{kuv'(k-u-v')}_{C,T,\partial S} + e^{kuv'(k-u-v')}_{T,C,\partial S}\right)$$

Therefore since u, v, u' and v' are dummy variables

$$I^k_{S,C,T}(x) = -\sum_{\substack{u+v=2\\u\neq 0\\v\neq 0}}^{k} e^{kuv(k-u-v)}_{T,C,\partial S}$$

Therefore we can rewrite (3.5) as

$$F^k_S(x) = F^k_C(x) + F^k_T(x) - \sum_{\substack{u+v=2\\u\neq 0\\v\neq 0}}^{k} e^{kuv(k-u-v)}_{T,C,\partial S}$$

Our aim is to decompose $\Delta F^k_S(x) \triangleq F^k_{S'}(x) - F^k_S(x)$ hence recall $F^k_{S'}(x)$

$$\begin{aligned}
F^k_{S'}(x) &= \sum_{\substack{u+v=k\\u\neq 0}} e^{kuv}_{S',\partial S} \\
&= \sum_{\substack{u+v=k\\u\neq 0}} \sum_{u'+v'=u} e^{ku'v'v}_{C',T',\partial S} \\
&= \sum_{\substack{u+v=k\\u\neq 0}} \{e^{ku0v}_{C',T',\partial S} + e^{k0uv}_{C',T',\partial S} + \sum_{\substack{u'+v'=u\\u'\neq 0\\v'\neq 0}} e^{ku'v'v}_{C',T',\partial S}\}
\end{aligned} \tag{3.6}$$

Once again for $k = u + v$

$$e^{kuv}_{C',\partial C} = \sum_{v'=0}^{v} e^{kuv'(k-u-v')}_{C',T,\partial S}$$

therefore

$$e^{ku0v}_{C',T,\partial S} = e^{kuv}_{C',\partial C} - \sum_{v'=1}^{v} e^{kuv'(k-u-v')}_{C',T,\partial S}$$

and

$$e^{ku0v}_{T',C,\partial S} = e^{kuv}_{T',\partial T} - \sum_{v'=1}^{v} e^{kuv'(k-u-v')}_{T',C,\partial S}$$

before using these in (3.6) note that $e^{ku0v}_{C',T,\partial S} = e^{ku0v}_{C',T',\partial S}$ and $e^{ku0v}_{T',C,\partial S} = e^{ku0v}_{T',C',\partial S}$, so we have

$$F^k_{S'}(x) = F^k_{C'}(x) + F^k_{T'}(x) + I^k_{S',C,T}(x) \tag{3.7}$$

where, skipping the exact index modifications we did before,

$$
\begin{aligned}
I^k_{S',C,T}(x) &= \sum_{\substack{u+v=k\\ u\neq 0}} \{ \sum_{\substack{u'+v'=u\\ u'\neq 0\\ v'\neq 0}} e^{ku'v'v}_{C',T',\partial S} - \sum_{v'=1}^{v} \left(e^{kuv'(k-u-v')}_{C',T,\partial S} + e^{kuv'(k-u-v')}_{T',C,\partial S} \right) \} \\
&= \sum_{\substack{u+v=2\\ u\neq 0\\ v\neq 0}}^{k} \left(e^{kuv(k-u-v)}_{C',T',\partial S} - e^{kuv(k-u-v)}_{C',T,\partial S} - e^{kuv(k-u-v)}_{T',C,\partial S} \right)
\end{aligned}
$$

Therefore combining (3.5) with (3.7) we have

$$
\Delta F^k_S(x) = \Delta F^k_C(x) + \Delta F^k_T(x) + \Delta I^k_{C,T}(x)
$$

hence

$$
\Delta F_S(x) = \Delta F_C(x) + \Delta F_T(x) + \Delta I_{C,T}(x) \tag{3.8}
$$

where

$$
\Delta I_{C,T}(x) = \sum_{k=2}^{K} \Delta I^k_{C,T}(x)
$$

and for any $k = 2, .., K$, the correction term $\Delta I^k_{C,T}(x) = I^k_{S',C,T}(x) - I^k_{S,C,T}(x)$ is given by

$$
\begin{aligned}
\Delta I^k_{C,T}(x) &= \sum_{\substack{u+v=2\\ u\neq 0\\ v\neq 0}}^{k} \left(e^{kuv(k-u-v)}_{T,C,\partial S} + e^{kuv(k-u-v)}_{C',T',\partial S} - e^{kuv(k-u-v)}_{C',T,\partial S} - e^{kuv(k-u-v)}_{T',C,\partial S} \right) \\
&= \sum_{\substack{u+v=2\\ u\neq 0\\ v\neq 0}}^{k} \left(e^{kuv(k-u-v)}_{C,T,\partial S} + e^{kuv(k-u-v)}_{C',T',\partial S} - e^{kuv(k-u-v)}_{C',T,\partial S} - e^{kuv(k-u-v)}_{C,T',\partial S} \right)
\end{aligned}
$$

Here last equation followed due to the symmetry between the indices.

Notice that any clique $\pi^\star$ involved in $\Delta I^k_{C,T}(x)$ is from $\Pi^{kuv(k-u-v)}_{C,T,\partial S}$ with positive u and v, therefore any pixel in $s \in \pi^\star \cap C$ belongs to ∂T and any pixel in $s \in \pi^\star \cap T$ belongs to ∂C. Hence the following must be clear.

$$
\Delta I^k_{C,T}(x) = \Delta I^k_{C\cap\partial T,T}(x) = \Delta I^k_{C,T\cap\partial C}(x) = \Delta I^k_{C\cap\partial T,T\cap\partial C}(x) \tag{3.9}
$$

and similarly

$$
\Delta I_{C,T}(x) = \Delta I_{C\cap\partial T,T}(x) = \Delta I_{C,T\cap\partial C}(x) = \Delta I_{C\cap\partial T,T\cap\partial C}(x) \tag{3.10}
$$

We now give the most important result of this section as it provides the core for the following sections.

Theorem 3.15 *Consider the MRF energy given by*

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x(\pi))$$

which is ***regular****, i.e., for any $\pi \subset \Sigma$, f_π is a regular function. Let $x \in \Lambda^\Sigma$ be an image, $S \subset \Sigma$ and $C_0 \subset S$ be a component with respect to x such that $\Delta F_U(x) \geq 0$ for any set $U \subset T_0 = \Sigma \setminus C_0$. Then there exists a component C with respect to x such that $C_0 \subset C \subset S$ and $\Delta F_C(x) \leq \Delta F_S(x)$, hence $F(\kappa_C \cdot x) \leq F(\kappa_S \cdot x)$.*

Proof Consider the following algorithm.

1. Set $i = 0$.

2. If $\Delta F_{C_i}(x) \leq \Delta F_S(x)$, set $C = C_i$ and terminate.

3. Pick a pixel $t \in \partial C_i \cap T_i$ such that $x(t) = x_{C_i}$. Notice that, since C_i is a component with this reconstruction, x_{C_i} is well defined. If no such t exists, set $C = C_i$ and terminate. Otherwise set $i = i + 1$.

4. Set $C_i = C_{i-1} \cup \{t\}$, $T_i = S \setminus C_i$. Goto step 2.

Notice that because of the way we expand it, the set C_i always remains a component with respect to x which includes C_0. It is evident that the algorithm terminates after $|T_0|$ steps at most. Let n denote the loop index i when the algorithm terminates. If the algorithm terminates at the second step, we are done. If it terminates at the third step we have two possibilities.

i. $T_n = \varnothing$, which means that S itself is a component and $C = C_n = S$ hence $\Delta F_C(x) = \Delta F_{C_n}(x) = \Delta F_S(x)$.

ii. there exists no pixel $t \in \partial C_n \cap T_n$ such that $x(t) = x_{C_n}$. Then, however, $x_{T_n \cap \partial C_n} = {x_{C_n}}'$ is well defined. Recall (3.8)

$$\Delta F_S^k(x) = \Delta F_{C_n}^k(x) + \Delta F_{T_n}^k(x) + \Delta I_{C_n,T_n}^k(x) \tag{3.11}$$

Recall (3.9) and denote $\bar{C} = C_n \cap \partial T_n$ and $\bar{T} = T_n \cap \partial C_n$, then

$$\begin{aligned}\Delta I^k_{C_n,T_n}(x) &= \Delta I^k_{\bar{C},\bar{T}}(x)\\ &= \sum_{\substack{u+v=2\\u\neq 0\\v\neq 0}}^{k} \left(e^{kuv(k-u-v)}_{\bar{C},\bar{T},\partial S} + e^{kuv(k-u-v)}_{\bar{C}',\bar{T}',\partial S} - e^{kuv(k-u-v)}_{\bar{C}',\bar{T},\partial S} - e^{kuv(k-u-v)}_{\bar{C},\bar{T}',\partial S}\right)\end{aligned}$$

Define

$$\Delta i^{kuv}_{\bar{C},\bar{T}}(x) = e^{kuv(k-u-v)}_{\bar{C},\bar{T},\partial S} + e^{kuv(k-u-v)}_{\bar{C}',\bar{T}',\partial S} - e^{kuv(k-u-v)}_{\bar{C}',\bar{T},\partial S} - e^{kuv(k-u-v)}_{\bar{C},\bar{T}',\partial S}$$

Recall from the Notation 3.13 that $\Delta i^{kuv}_{\bar{C},\bar{T}}(x)$ is given by

$$\sum_{\pi\in\Pi^{kuv(k-u-v)}_{\bar{C},\bar{T},\partial S}} f_\pi(x(\pi)) + f_\pi(\kappa_{\bar{C}\bar{T}}(x)(\pi)) - f_\pi(\kappa_{\bar{C}}(x)(\pi)) - f_\pi(\kappa_{\bar{T}}(x)(\pi)) \quad (3.12)$$

For any clique $\pi \in \Pi^{kuv(k-u-v)}_{\bar{C},\bar{T},\partial S}$, define

$$\begin{aligned}\alpha^1 &= \{j \in I_k : \pi_j \in \bar{C}\}\\ \alpha^2 &= \{j \in I_k : \pi_j \in \bar{T}\}\end{aligned}$$

let $\alpha^0 = I_k \setminus (\bar{C} \cup \bar{T})$ and denote $a_j = x_{\pi_{\alpha^0_j}}$ for $j = 1, .., k-u-v$. Observe that

$$\mathcal{R}^{(a_1..a_{k-u-v})}_{\alpha^0;\alpha^1,\alpha^2}(f_\pi) = f_\pi(x(\pi)) + f_\pi(\kappa_{\bar{C}\bar{T}}(x)(\pi)) - f_\pi(\kappa_{\bar{C}}(x)(\pi)) - f_\pi(\kappa_{\bar{T}}(x)(\pi))$$

is nonnegative by the hypothesis. However this and (3.12) imply that $\Delta i^{kuv}_{\bar{C},\bar{T}}(x)$, $\Delta I^k_{C_n,T_n}(x)$ and $\Delta I_{C_n,T_n}(x)$ are nonnegative as well. Recalling (3.11) and the fact that $\Delta F_{T_n}(x) \geq 0$ by the hypothesis, this proves $\Delta F^k_{C_n}(x) \leq \Delta F^k_S(x)$ and completes the proof. $\square$

The following example demonstrates how we are going to use this theorem in the sequel.

Example 3.16 Let an image $x^\star$ be a minimizer of

$$F(x) = \sum_{k=1}^{K} \sum_{\pi\in\Pi^k} f_\pi(x(\pi))$$

where, for any clique $\pi \subset \Sigma$ f_π is regular. Pick a 1-clique (or pixel) $\pi^\star \subset \Sigma$ and define

$$\hat{F}(x) = F(x) - f_{\pi^\star}(x(\pi^\star)) + \hat{f}_{\pi^\star}(x(\pi^\star))$$

where $\hat{f}_{\pi^\star} \in V_L^1$. Assume that $x^\star$ is not a minimizer of $\hat{F}$. Then our aim is to find a minimizer $\hat{x}^\star$ of $\hat{F}$ starting from $x^\star$. Hence we look for a *minimizing set* $\hat{S}$, i.e.

$$\hat{S} = arg \min_{S \subset \Sigma} \Delta \hat{F}_S(x^\star)$$

to assign $\hat{x}^\star = \kappa_{\hat{S}} \cdot x^\star$. Since $x^\star$ is not a minimizer of $\hat{F}$ we have $\hat{S} \neq \varnothing$. Moreover since $x^\star$ is a minimizer of F, therefore

$$\Delta \hat{F}_T(x^\star) = \Delta F_T(x^\star) \geq 0$$

for any set $T \subset \Sigma \backslash \pi^\star$, we need to have $\pi^\star \subset \hat{S}$. Hence Theorem 3.15 asserts that $\hat{S}$ should be a component with respect to $x^\star$. This assertion is going to prove to be very useful in the following section. □

3.2.3 Monotonicity of Minimizers

The monotonicity property of minimizers of binary MRFs is going to be the building block of our algorithm. Before introducing it we need to present yet another notation.

Definition 3.17 *Let $f, g \in V_2^n$ be two functions. We denote $g \geq f$, if for any $(p; q)$ partition $\{\alpha^0; \alpha^1\}$ of I_n and for any $a = (a_1, .., a_p) \in \Lambda_2^p$ we have*

$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(g)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(g)(1) \geq \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(f)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(f)(1) \tag{3.13}$$

If neither $g \geq f$ nor $f \geq g$ holds, f and g are said to be ***incomparable****.*

Example 3.18 Let $f, g \in V_2^2$ for which $g \geq f$ holds. Thence (3.13) amounts to

$$g(0,0) - g(0,1) \geq f(0,0) - f(0,1)$$

for $\alpha^0 = \{1\}$, $\alpha^1 = \{2\}$ and $a = (0)$, or

$$g(0,0) - g(1,1) \geq f(0,0) - f(1,1)$$

for $\alpha^0 = \varnothing$ and $\alpha^1 = \{1, 2\}$. □

For solely the sake of completeness we state the following proposition which asserts that the relation $\geq$ defined above is a partial ordering in V_2^n. We leave the proof to Appendix B.

Proposition 3.19 *The family V_2^n with the relation $\geq$ given by Definition 3.17 is a partially ordered set, i.e.*

$$\begin{aligned} i. &\quad f \geq f \\ ii. &\quad g \geq f \text{ and } f \geq g \implies g = f \\ iii. &\quad g \geq f \text{ and } f \geq h \implies g \geq h \end{aligned}$$

for any $f, g, h \in V_2^n$.

We leave the proof of the following to Appendix C.

Proposition 3.20 *Let $f, g \in V_2^n$. If for any $(n-1;1)$ partition $\{\alpha^0;\alpha^1\}$ of I_n and for any $a = (a_1, .., a_{n-1}) \in \Lambda_2^{n-1}$ we have*

$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(g)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(g)(1) \geq \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(f)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(f)(1)$$

then $g \geq f$.

Definition 3.21 *Let $x, y \in \Lambda_2^\Sigma$ be two images. We say $x \geq y$, if for any $i \in \Sigma$ we have $x_i \geq y_i$. If neither $x \geq y$ nor $y \geq x$ holds, images x and y are said to be **incomparable**.*

Notice that Λ_2^Σ is partially ordered with $\geq$.

Proposition 3.22 *Let $f, g \in V_2^n$ be two functions, then $g \geq f$ if and only if for any $x, y \in \Lambda_2^n$ such that $x \leq y$ we have*

$$g(x) - g(y) \geq f(x) - f(y)$$

Proof Assume $g \geq f$ and let $x, y \in \Lambda_2^n$ be any two n-tuples such that $x \leq y$. Define a partition $\{\alpha^0;\alpha^1\}$ of I_n as follows

$$\alpha^0 = \{\alpha_1^0, .., \alpha_p^0\} = \{i \in I_n : x_i = y_i\}$$
$$\alpha^1 = I_n \setminus \alpha^0$$

and define $a_j = x_{\alpha^0_j} = y_{\alpha^0_j}$ for $j = 1, .., p$ where p is the number of indices for which $x_i = y_i$, $i = 1, .., n$. Then certainly

$$f(x) = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(0) \qquad g(x) = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(0)$$
$$f(y) = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(1) \qquad g(y) = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(1)$$

hence $g \geq f$ implies

$$g(x) - g(y) \geq f(x) - f(y)$$

Conversely let $\{\alpha^0; \alpha^1\}$ be a $(p; q)$ partition of I_n and let $a = (a_1, .., a_p) \in \Lambda_2^p$. Define $x = (x_1, .., x_n)$ and $y = (y_1, .., y_n)$ as follows

$$\begin{aligned} x_{\alpha^0_j} = y_{\alpha^0_j} = a_j, &\quad \text{for } j = 1, .., p \\ x_{\alpha^1_j} = 0, &\quad \text{for } j = 1, .., q \\ y_{\alpha^1_j} = 1, &\quad \text{for } j = 1, .., q \end{aligned}$$

Certainly $x \leq y$ and

$$g(x) - g(y) \geq f(x) - f(y)$$

implies that

$$\mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(1) \geq \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(1)$$

and completes the proof. $\square$

Recall the MRF energy function

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

where for any clique $\pi \subset \Sigma$, f_π is regular. Let $x^\star \in \Lambda_2^\Sigma$ be a minimizer of F. Let us slightly alter the MRF energy and define

$$G(x) = F(x) - f_{\pi^\star}(x_{\pi^\star}) + g_{\pi^\star}(x_{\pi^\star}) \tag{3.14}$$

where $\pi^\star \subset \Sigma$ is an n-clique and $g_{\pi^\star} \in V_2^n$ is a regular function. Our aim is to find a minimizer of G in terms of $x^\star$.

First observe that either $x^\star$ is a minimizer of G or there should exist a subset S of Σ such that $\kappa_S \cdot x^\star$ minimizes G. Note that for any image $x \in \Lambda_2^\Sigma$

$$\begin{aligned}
G(x^\star) &= F(x^\star) - f_{\pi^\star}(x^\star_{\pi^\star}) + g_{\pi^\star}(x^\star_{\pi^\star}) \\
&\leq F(x) - f_{\pi^\star}(x^\star_{\pi^\star}) + g_{\pi^\star}(x^\star_{\pi^\star}) \\
&= G(x) - g_{\pi^\star}(x_{\pi^\star}) + f_{\pi^\star}(x_{\pi^\star}) - f_{\pi^\star}(x^\star_{\pi^\star}) + g_{\pi^\star}(x^\star_{\pi^\star}) \\
&= G(x) + (f_{\pi^\star}(x_{\pi^\star}) - f_{\pi^\star}(x^\star_{\pi^\star})) - (g_{\pi^\star}(x_{\pi^\star}) - g_{\pi^\star}(x^\star_{\pi^\star})) \qquad (3.15)
\end{aligned}$$

Note that for both of the following cases

$$\begin{aligned}
&i. \quad g_{\pi^\star} \geq f_{\pi^\star} \quad \text{and} \quad x_{\pi^\star} \leq x^\star_{\pi^\star} \\
&ii. \quad f_{\pi^\star} \geq g_{\pi^\star} \quad \text{and} \quad x^\star_{\pi^\star} \leq x_{\pi^\star}
\end{aligned}$$

we have, due to the Proposition 3.22, that

$$g_{\pi^\star}(x_{\pi^\star}) - g_{\pi^\star}(x^\star_{\pi^\star}) \geq f_{\pi^\star}(x_{\pi^\star}) - f_{\pi^\star}(x^\star_{\pi^\star})$$

However, using this in (3.15) gives

$$G(x^\star) \leq G(x)$$

hence shows that there should exist a minimizer $y^\star$ of G such that either $y^\star$ and $x^\star$ are incomparable or either of the following holds.

$$\begin{aligned}
&i. \quad g_{\pi^\star} \geq f_{\pi^\star} \text{ and } y^\star \geq x^\star \\
&ii. \quad g_{\pi^\star} \leq f_{\pi^\star} \text{ and } y^\star \leq x^\star
\end{aligned} \qquad (3.16)$$

Our next aim is to prove that for any minimizer $x^\star$ of F there exists a minimizer of G which is not incomparable with $x^\star$.

This is indeed quite straightforward if $\pi^\star$ is a 1-clique. If $\pi^\star$ is a 1-clique and if $x^\star$ is not a minimizer of G there should exist some subset S of Σ, for which $\pi^\star \in S$, so that $\kappa_S \cdot x^\star$ minimizes G. Note that for any $U \subset S \setminus \pi^\star$, $\Delta G_U(x^\star) = \Delta F_U(x^\star) \geq 0$ since $x^\star$ is a minimizer of F. Then, however, as Theorem 3.15 asserts, there exists a component C such that $\pi^\star \subset C \subset S$ and

$$G(\kappa_C \cdot x^\star) \leq G(\kappa_S \cdot x^\star)$$

This proves that, $y^\star = \kappa_C \cdot x^\star$, which is not incomparable with $x^\star$, minimizes G. Using this argument along with (3.16) as many times as needed proves the following.

Theorem 3.23 *Consider the MRF energy functions*

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$
$$G(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} g_\pi(x_{\pi_1}, .., x_{\pi_k})$$

where for any clique $\pi \subset \Sigma$, f_π and g_π are regular functions and

$$\begin{aligned} g_\pi &\geq (\leq) f_\pi, \quad \textit{for } \pi \in \Pi^1 \\ g_\pi &= f_\pi, \quad \textit{for } \pi \in \Pi^k,\ k > 1 \end{aligned} \tag{3.17}$$

holds. Then for any minimizer $x^\star$ of F, there exists a minimizer $y^\star$ of G for which $y^\star \geq (\leq) x^\star$ holds.

This is what we call the monotonicity property of minimizers of binary MRFs. This property also appears in [10, 8, 33, 18] for $K = 2$. Our presentation provides an obvious generalization and abstraction. In the rest of this section we are going further generalize this result by relaxing Equation (3.17). Before doing that however we need to explore a few more properties of minimizers of F. Again we assume that $x^\star$ is a minimizer of F and $\pi^\star$ is an n-clique where $n > 1$.

Consider the energy $F(\bar{x})$ where $\bar{x}$ is a *constrained variable* for which $\bar{x}_{\pi^\star} \geq x_{\pi^\star}$ holds. We are going to show that there exists a minimizer $\bar{x}^\star$ for the energy $F(\bar{x})$ such that $\bar{x}^\star \geq x^\star$ holds. Similarly for the energy $F(\tilde{x})$ where $\tilde{x}$ is constrained such that $\tilde{x}_{\pi^\star} \leq x_{\pi^\star}$, we are going to prove that there exists a minimizer $\tilde{x}^\star$ such that $\tilde{x}^\star \leq x^\star$ holds. To prove it, we are not going to use the constrained variables though, instead we are going to employ a different approach.

Let $\alpha^0 = \{\alpha^0_1, .., \alpha^0_p\} \subset I_n$ be an arbitrary set of indices. Define $\alpha^1 = \{\alpha^1_1, .., \alpha^1_{q_1}\} \subset (I_n \setminus \alpha^0)$ and $\alpha^2 = \{\alpha^2_1, .., \alpha^2_{q_2}\} \subset (I_n \setminus \alpha^0)$ so that $\{\alpha^0; \alpha^1, \alpha^2\}$

is a $(p; q_1, q_2)$ partition of I_n and the following holds.

$$x^\star_{\pi^\star_{\alpha^1_i}} = 0, \quad \text{for } 1 \le i \le q_1$$
$$x^\star_{\pi^\star_{\alpha^2_i}} = 1, \quad \text{for } 1 \le i \le q_2$$

Define $a = (a_1, .., a_n) \in \Lambda_2^n$, $b = (b_1, .., b_n) \in \Lambda_2^n$ and $c = (c_1, .., c_n) \in \Lambda_2^n$ such that

$$\begin{aligned} a_{\alpha^0_j} = b_{\alpha^0_j} = c_{\alpha^0_j} = x^\star_{\pi^\star_{\alpha^0_j}}, &\quad \text{for } 1 \le j \le p \\ a'_{\alpha^1_j} = b'_{\alpha^1_j} = c_{\alpha^1_j} = x^\star_{\pi^\star_{\alpha^1_j}} = 0, &\quad \text{for } 1 \le j \le q_1 \\ a'_{\alpha^2_j} = b_{\alpha^2_j} = c'_{\alpha^2_j} = x^\star_{\pi^\star_{\alpha^2_j}} = 1, &\quad \text{for } 1 \le j \le q_2 \end{aligned}$$

Notice that $b \ge a \ge c$, $b \ge x^\star_{\pi^\star} \ge c$ and that $x^\star_{\pi^\star}$ and a are not comparable unless $x^\star_{\pi^\star} = a$ which holds when $q_1 = q_2 = 0$. For the following we assume $q_1 + q_2 > 0$.

Example 3.24 Let $\pi^\star$ be a 5-clique, $x^\star_{\pi^\star} = (0, 1, 0, 0, 1)$ and $\alpha^0 = \{2, 3\}$. Therefore $\alpha^1 = \{1, 4\}$ and $\alpha^2 = \{5\}$. Then $a = (1, 1, 0, 1, 0)$, $b = (1, 1, 0, 1, 1)$ and $c = (0, 1, 0, 0, 0)$. □

View each pixel $\pi^\star_i$, $i \in I_n$, as a 1-clique in Σ and for each $j = 1, .., p$ define functions of single variable $\hat{f}_{\pi^\star_{\alpha^0_j}}$ as

$$\hat{f}_{\pi^\star_{\alpha^0_j}}(x^\star_{\pi^\star_{\alpha^0_j}}{}') = \infty$$
$$\hat{f}_{\pi^\star_{\alpha^0_j}}(x^\star_\pi) = f_{\pi^\star_{\alpha^0_j}}(x^\star_\pi)$$

Here ∞ denotes a sufficiently large number. Since, for any clique $\pi \subset \Sigma$, f_π is defined to be finite and Σ is defined to be a finite set, as we progress, it is going to be clear that this definition of ∞ is unambiguous. The reason why we cannot use the conventional *infinity* is the first requirement in Definition 2.10. Define

$$\hat{F}(x) = F(x) - \sum_{i \in \alpha^0} f_{\pi^\star_i}(x_{\pi^\star_i}) + \sum_{i \in \alpha^0} \hat{f}_{\pi^\star_i}(x_{\pi^\star_i})$$

Since $\hat{F}(x^\star) = F(x^\star) \le F(x) \le \hat{F}(x)$ for any image x, $x^\star$ is a minimizer of $\hat{F}$.

For each $j = 1, .., q_1$ define functions of single variable $\bar{f}_{\pi^\star_{\alpha^1_j}}$ as

$$\bar{f}_{\pi^\star_{\alpha^1_j}}(0) = \infty$$

$$\bar{f}_{\pi^\star_{\alpha^1_j}}(1) = f_{\pi^\star_{\alpha^1_j}}(1)$$

Construct the following MRF energy

$$\bar{F}(x) = \hat{F}(x) - \sum_{i \in \alpha^1} f_{\pi^\star_i}(x_{\pi^\star_i}) + \sum_{i \in \alpha^1} \bar{f}_{\pi^\star_i}(x_{\pi^\star_i})$$

Note that for any $\pi \in \Pi^1$ we have

$$\bar{f}_\pi(0) - \bar{f}_\pi(1) \geq \hat{f}_\pi(0) - \hat{f}_\pi(1)$$

therefore $\bar{f}_\pi \geq \hat{f}_\pi$ and for any $\pi \in \Pi^k$, with $k > 1$, we have $\bar{f}_\pi = \hat{f}_\pi$. Thus we know from Theorem 3.23 that there exists a minimizer $\bar{x}^\star$ of $\bar{F}$ for which $\bar{x}^\star \geq x^\star$. Notice moreover that

$$\bar{x}^\star = arg \min_{x_{\pi^\star} = b} F(x)$$

Similarly defining functions of single variable $\tilde{f}_{\pi^\star_{\alpha^2_j}}$ as

$$\tilde{f}_{\pi^\star_{\alpha^2_j}}(1) = \infty$$

$$\tilde{f}_{\pi^\star_{\alpha^2_j}}(0) = f_{\pi^\star_{\alpha^2_j}}(0)$$

for each $j = 1, .., q_2$ we construct the MRF energy

$$\tilde{F}(x) = \hat{F}(x) - \sum_{i \in \alpha^2} f_{\pi^\star_i}(x_{\pi^\star_i}) + \sum_{i \in \alpha^2} \tilde{f}_{\pi^\star_i}(x_{\pi^\star_i})$$

Observe that for any $\pi \in \Pi^1$ we have

$$\tilde{f}_\pi(0) - \tilde{f}_\pi(1) \leq \hat{f}_\pi(0) - \hat{f}_\pi(1)$$

therefore $\tilde{f}_\pi \leq \hat{f}_\pi$ and for any $\pi \in \Pi^k$, with $k > 1$, we have $\tilde{f}_\pi = \hat{f}_\pi$. Hence another call to Theorem 3.23 asserts that there exists a minimizer $\tilde{x}^\star$ of $\tilde{F}$ for which $\tilde{x}^\star \leq x^\star$ and

$$\tilde{x}^\star = arg \min_{x_{\pi^\star} = c} F(x)$$

Since the set α^0 was arbitrary, we proved

Lemma 3.25 *Consider the MRF energy function*

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

where for any clique $\pi \subset \Sigma$, f_π is regular. Let $\pi^\star \subset \Sigma$ be an n-clique and $x^\star \in \Lambda_2^\Sigma$ be a minimizer of F. Let $b, c \in \Lambda_2^n$ for which $b \geq x^\star_{\pi^\star} \geq c$ hold. Then there exist images $\bar{x}^\star$ and $\tilde{x}^\star$ so that

$$\bar{x}^\star = arg \min_{x_{\pi^\star}=b} F(x)$$

$$\tilde{x}^\star = arg \min_{x_{\pi^\star}=c} F(x)$$

and $\bar{x}^\star \geq x^\star \geq \tilde{x}^\star$.

Next we show that $F(\bar{x}^\star) \leq F(\check{x}^\star)$ and $F(\tilde{x}^\star) \leq F(\check{x}^\star)$, where $\check{x}^\star$ denotes the minimizer of $F(\check{x})$ where $\check{x}$ is constrained to satisfy the condition that $\check{x}_{\pi^\star} = a$.

Construct the following MRF energy

$$\check{F}(x) = \bar{F}(x) - \sum_{i \in \alpha^2} f_{\pi_i^\star}(x_{\pi_i^\star}) + \sum_{i \in \alpha^2} \tilde{f}_{\pi_i^\star}(x_{\pi_i^\star})$$

Note that for any minimizer $\check{x}^\star$ of $\check{F}$ we have

$$\check{x}^\star = arg \min_{x_{\pi^\star}=a} F(x)$$

However

$$\min \check{F}(x) \geq \min \bar{F}(x) = \min_{x_{\pi^\star}=b} F(x)$$

Thus

$$\min_{x_{\pi^\star}=b} F(x) \leq \min_{x_{\pi^\star}=a} F(x) \tag{3.18}$$

Similarly, we also have

$$\check{F}(x) = \tilde{F}(x) - \sum_{i \in \alpha^1} f_{\pi_i^\star}(x_{\pi_i^\star}) + \sum_{i \in \alpha^1} \bar{f}_{\pi_i^\star}(x_{\pi_i^\star})$$

Since

$$\min \check{F}(x) \geq \min \tilde{F}(x) = \min_{x_{\pi^\star}=c} F(x)$$

We similarly end up with

$$\min_{x_{\pi^\star}=c} F(x) \leq \min_{x_{\pi^\star}=a} F(x) \tag{3.19}$$

Now define

$$G(x) = F(x) - f_{\pi^\star}(x_{\pi^\star}) + g_{\pi^\star}(x_{\pi^\star}) \tag{3.20}$$

where $g_{\pi^\star} \in V_2^n$ is a regular function. Note that by this definition

$$\check{x}^\star = arg \min_{x_{\pi^\star}=a} G(x)$$

Let $y \in \Lambda_2^\Sigma$ be an image so that $y_{\pi^\star} = a$, then

$$\begin{aligned} G(y) &\geq G(\check{x}^\star) \\ &= F(\check{x}^\star) - f_{\pi^\star}(a) + g_{\pi^\star}(a) \end{aligned} \tag{3.21}$$

Let $g_{\pi^\star} \geq f_{\pi^\star}$, then due to the Proposition 3.22 and the fact that $b \geq a$ we have

$$g_{\pi^\star}(a) - g_{\pi^\star}(b) \geq f_{\pi^\star}(a) - f_{\pi^\star}(b)$$

using this and (3.18) in (3.21) gives

$$\begin{aligned} G(y) &\geq F(\bar{x}^\star) - f_{\pi^\star}(b) + g_{\pi^\star}(b) \\ &= G(\bar{x}^\star) \end{aligned} \tag{3.22}$$

On the other hand if $g_{\pi^\star} \leq f_{\pi^\star}$, since $a \geq c$ we have

$$f_{\pi^\star}(c) - f_{\pi^\star}(a) \geq g_{\pi^\star}(c) - g_{\pi^\star}(a)$$

due to the Proposition 3.22. Using this and (3.19) in (3.21) gives

$$\begin{aligned} G(y) &\geq F(\tilde{x}^\star) - f_{\pi^\star}(c) + g_{\pi^\star}(c) \\ &= G(\tilde{x}^\star) \end{aligned} \tag{3.23}$$

Note that since the set α^0 was arbitrary, the inequalities (3.22) and (3.23) prove that there exists a minimizer $\hat{x}^\star$ of G such that

$$i. \quad \hat{x}^\star_{\pi^\star} \geq x^\star_{\pi^\star} \text{ if } g_{\pi^\star} \geq f_{\pi^\star}$$
$$ii. \quad \hat{x}^\star_{\pi^\star} \leq x^\star_{\pi^\star} \text{ if } g_{\pi^\star} \leq f_{\pi^\star}$$

hold.

So assume that $g_{\pi^\star}$ and $f_{\pi^\star}$ are comparable so that there exists a minimizer $\hat{x}^\star$ of G such that $\hat{x}^\star_{\pi^\star}$ and $x^\star_{\pi^\star}$ are comparable. Since, by definition

$$\hat{x}^\star = arg \min_{x_{\pi^\star}=\hat{x}^\star_{\pi^\star}} G(x)$$

Equation (3.20) implies that

$$\hat{x}^\star = arg \min_{x_{\pi^\star}=\hat{x}^\star_{\pi^\star}} F(x)$$

However thanks to Lemma 3.25 we know that there exists a minimizer $y^\star$ of F which not only has the property that

$$y^\star = arg \min_{x_{\pi^\star}=\hat{x}^\star_{\pi^\star}} F(x) = arg \min_{x_{\pi^\star}=\hat{x}^\star_{\pi^\star}} G(x)$$

but also is comparable with $x^\star$. Arguing the same way as many times as necessary we finally proved

Theorem 3.26 *Consider the MRF energy functions*

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$
$$G(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} g_\pi(x_{\pi_1}, .., x_{\pi_k})$$

where for any clique $\pi \subset \Sigma$, f_π and g_π are regular functions and $g_\pi \geq (\leq) f_\pi$. Then for any minimizer $x^\star$ of F, there exists a minimizer $y^\star$ of G for which $y^\star \geq (\leq) x^\star$ holds.

3.2.4 Extensions

We finally present our algorithm to minimize MRF energies in this section. The algorithm is going to be evident once we extend the results of previous section to L-ary MRFs.

Notation 3.27 *For any $x \in \Lambda_L$ and $j = 0,..,L-2$ we denote*

$$x^j = \begin{cases} 1 & : \quad j < x \\ 0 & : \quad j \geq x \end{cases}$$

Notice that for $x \in \Lambda_L$, we have $x = \sum_{i=0}^{L-2} x^i$ and $x^i \geq x^j$ if $i \leq j$. Conversely for any $(a_0,..,a_{L-2}) \in \Lambda_2^{L-2}$, for which $a_i \geq a_j$ when $i \leq j$, we have an integer $a \in \Lambda_L$ for which $a = \sum_{i=0}^{L-2} a_i$ and $a_i = a^i$ holds.

Definition 3.28 *Let $x = (x_1,..,x_M) \in \Lambda_L^\Sigma$ be an L-**ary image**. We call the binary image $x^i = (x_1^i,..,x_M^i) \in \Lambda_2^\Sigma$, the i-**th level set of** x.*

Note that by this definition we have $x^i \geq x^j$ whenever $i \leq j$ for any image x.

Definition 3.29 *A function $f \in V_L^n$ is called **levelable** if*

$$f(x_1,..,x_n) = \sum_{i=0}^{L-2} f^i(x_1^i,..,x_n^i) \tag{3.24}$$

where for any nonnegative $i, j \leq L-2$,

$$i. \quad f^i \in V_2^n \text{ is regular} \tag{H1}$$

$$ii. \quad f^i \geq f^j \text{ if } i \leq j \tag{H2}$$

Example 3.30 Let $C \in \Lambda_L$ and let $f \in V_L^n$ be defined as

$$f(x_1,..,x_n) = c \cdot max(C, x_1,..,x_n) - c \cdot C$$

for some nonnegative $c \in \mathbb{R}$. We obviously have

$$f(x_1,..,x_n) = \sum_{i=0}^{L-2} f^i(x_1^i,..,x_n^i)$$

where $f^i \in V_2^n$ for $0 \leq i \leq L-2$ is defined as

$$f^i(a_1,..,a_n) = c \cdot max(C^i, a_1,..,a_n) - c \cdot C^i$$

for any $a_1,..,a_n \in \Lambda_2$. Note that $f^i(a_1,..,a_n) = 0$ when $C^i = 1$ and otherwise $f^i(a_1,..,a_n) = c \cdot \max(a_1,..,a_n)$. Therefore for any $(p; q_1, q_2)$ partition $\{\alpha_0; \alpha_1, \alpha_2\}$ of I_n and any sequence of binary constants $a_1,..,a_p$ we have

$$\mathcal{P}_{\alpha_0;\alpha_1,\alpha_2}^{(a_1..a_p)}(f^i)(0,1) = \mathcal{P}_{\alpha_0;\alpha_1,\alpha_2}^{(a_1..a_p)}(f^i)(1,0) = \mathcal{P}_{\alpha_0;\alpha_1,\alpha_2}^{(a_1..a_p)}(f^i)(1,1) = \mathcal{P}_{\alpha_0;\alpha_1,\alpha_2}^{(a_1..a_p)}(f^i)(0,0)$$

if $C^i = 1$ and

$$\mathcal{P}^{(a_1..a_p)}_{\alpha_0;\alpha_1,\alpha_2}(f^i)(0,1) = \mathcal{P}^{(a_1..a_p)}_{\alpha_0;\alpha_1,\alpha_2}(f^i)(1,0) = \mathcal{P}^{(a_1..a_p)}_{\alpha_0;\alpha_1,\alpha_2}(f^i)(1,1) \geq \mathcal{P}^{(a_1..a_p)}_{\alpha_0;\alpha_1,\alpha_2}(f^i)(0,0)$$

otherwise. This verifies that f^i is regular. To check H2, let $i \geq j$ and note that $f^i = f^j$ if $C^i = C^j$. On the other hand $C^i \neq C^j$ implies that $1 = C^j \geq C^i = 0$. Let $\{\alpha; \beta\}$ be a $(p;q)$ partition of I_n and let $a_1, .., a_p$ be a given set of binary numbers. Then we have

$$\begin{aligned}
&\mathcal{P}^{(a_1..a_p)}_{\alpha;\beta}(f^j)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha;\beta}(f^j)(1) = 0 \\
&\mathcal{P}^{(a_1..a_p)}_{\alpha;\beta}(f^i)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha;\beta}(f^i)(1) = c \cdot max(a_1, .., a_p) - c \leq 0
\end{aligned}$$

hence $f^j \geq f^i$ holds. This proves that f is a levelable function.

A very similar argument shows that $f \in V_L^n$ defined as

$$\begin{aligned}
f(x_1, .., x_n) &= -c \cdot min(C, x_1, .., x_n) \\
&= \sum_{i=0}^{L-2} -c \cdot min(C^i, x_1^i, .., x_n^i)
\end{aligned}$$

for some $C \in \Lambda_L$ and nonnegative $c \in \mathbb{R}$ is also a levelable function. $\square$

For functions of single variable, levelability is equivalent to convexity as we show next.

Proposition 3.31 *Let $f \in V_L^1$ be a convex function, i.e.*

$$2f(x) \leq f(x+1) + f(x-1)$$

for $1 \leq x \leq L-2$, then f is levelable.

Proof Note that for any $f \in V_L^1$ and any $x \in \Lambda_L$ the following holds.

$$f(x) = \sum_{i=0}^{L-2} (f(i+1) - f(i))x^i$$

Recall that $f(0) = 0$. For any nonnegative $i \leq L-2$, define $f^i \in V_2^1$ as

$$f^i(a) = (f(i+1) - f(i))a$$

for $a \in \Lambda_2$. Since any function of single variable is regular, f^i is a regular function. Since f is convex on the other hand, for $1 \le i \le L-2$ we have

$$f^{i-1}(1) = f(i) - f(i-1) \le f(i+1) - f(i) = f^i(1)$$

however as $f^i(0) = 0$ for any $0 \le i \le L-2$

$$f^{i-1}(0) - f^{i-1}(1) \ge f^i(0) - f^i(1)$$

which in turn means that $f^{i-1} \ge f^i$. This verifies H2 and proves the claim. $\square$

The following proposition is quite obvious.

Proposition 3.32 *Sum of levelable functions is levelable.*

Example 3.33 Let $f_1, .., f_n \in V_L^1$ be levelable functions, then the function $f \in V_L^n$ defined by

$$f(x_1, .., x_n) = f_1(x_1) + .. + f_n(x_n)$$

is levelable. For example for some $C \in \Lambda_L$ and nonnegative constants $c_1, .., c_n \in \mathbb{R}$

$$f(x_1, .., x_n) = c_1(x_1 - C)^2 + .. + c_n(x_n - C)^2 - nC^2$$

is a levelable function. $\square$

Recall the MRF energy function

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x_{\pi_1}, .., x_{\pi_k})$$

and assume that F is **levelable**, i.e for any $\pi \in \Pi^k$, $f_\pi \in V_L^k$ is a levelable function. We have

$$\begin{aligned}
F(x) &= \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} \sum_{i=0}^{L-2} f_\pi^i(x_{\pi_1}^i, .., x_{\pi_k}^i) \\
&= \sum_{i=0}^{L-2} \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi^i(x_{\pi_1}^i, .., x_{\pi_k}^i) \\
&= \sum_{i=0}^{L-2} F^i(x^i)
\end{aligned}$$

where each F^i is a binary MRF energy. Our aim is simply to minimize each F^i to obtain $x^{\star i}$ and construct the minimizer $x^\star$ of F out of $x^{\star 0}, .., x^{\star L-2}$ by adding them up, i.e

$$x^\star_s = \sum_{i=0}^{L-2} x^{\star i}_s$$

for any $s \in \Sigma$. However this procedure applies only if the collection of binary minimizers constitute a monotone sequence of minimizers, i.e.

$$x^{\star j} \geq x^{\star i} \text{ when } i \geq j \tag{3.25}$$

Thanks to Theorem 3.26, we know that there exist minimizers $x^{\star 0}, .., x^{\star L-2}$ of $F^0, .., F^{L-2}$ respectively, which satisfy the monotonicity condition (3.25). Hence an algorithm reveals itself; we pick an integer i, such that $0 \leq i \leq L-2$, we minimize F^i to obtain $x^{\star i}$, we then find out other minimizers making sure that the monotonicity condition (3.25) holds. Note that we are free to select the first level i and any of the minimizers of F^i, however for any other level j we need to pick an appropriate minimizer of F^j so that monotonicity is not broken. Note moreover that we need to perform $L-1$ binary MRF minimizations for this algorithm.

Example 3.34 Let $L = 4$ and $\Sigma = \{1, 2, 3\}$. Presume the trivial neighborhood system hence any subset of Σ is a clique. Define, for first order cliques

$$f_{\{1\}}(x_1) = (x_1 - 3)^2 - 9$$
$$f_{\{2\}}(x_2) = (x_2)^2$$
$$f_{\{3\}}(x_3) = (x_3 - 1)^2 - 1$$

for second order cliques

$$f_{\{1,2\}}(x_1, x_2) = max(3, x_1, x_2) - min(3, x_1, x_2) - 3$$
$$f_{\{2,3\}}(x_2, x_3) = f_{\{1,3\}}(x_1, x_3) = 0$$

and for the third order clique

$$f_{\{1,2,3\}}(x_1, x_2, x_3) = max(x_1, x_2, x_3) - min(x_1, x_2, x_3)$$

Define the MRF energy

$$F(x) = f_{\{1\}}(x_1) + f_{\{2\}}(x_2) + f_{\{3\}}(x_3) + f_{\{1,2\}}(x_1, x_2) + f_{\{1,2,3\}}(x_1, x_2, x_3)$$

Then we have

$$F^i(x^i) = f^i_{\{1\}}(x^i_1) + f^i_{\{2\}}(x^i_2) + f^i_{\{3\}}(x^i_3) + f^i_{\{1,2\}}(x^i_1, x^i_2) + f^i_{\{1,2,3\}}(x^i_1, x^i_2, x^i_3)$$

where for first order cliques

$$\begin{aligned}
f^i_{\{1\}}(x^i_1) &= [f_{\{1\}}(i+1) - f_{\{1\}}(i)]x^i_1 = [(i+1-3)^2 - (i-3)^2]x^i_1 \\
f^i_{\{2\}}(x^i_2) &= [f_{\{2\}}(i+1) - f_{\{1\}}(i)]x^i_2 = [(i+1)^2 - i^2]x^i_2 \\
f^i_{\{3\}}(x^i_3) &= [f_{\{3\}}(i+1) - f_{\{3\}}(i)]x^i_3 = [(i+1-1)^2 - (i-1)^2]x^i_3
\end{aligned}$$

for second order cliques

$$\begin{aligned}
f^i_{\{1,2\}}(x^i_1, x^i_2) &= max(3^i, x^i_1, x^i_2) - min(3^i, x^i_1, x^i_2) - 3^i \\
&= max(1, x^i_1, x^i_2) - min(1, x^i_1, x^i_2) - 1
\end{aligned}$$

and $f^i_{\{2,3\}}(x^i_2, x^i_3) = f^i_{\{1,3\}}(x^i_1, x^i_3) = 0$, for the third order clique

$$f^i_{\{1,2,3\}}(x^i_1, x^i_2, x^i_3) = max(x^i_1, x^i_2, x^i_3) - min(x^i_1, x^i_2, x^i_3)$$

Thus denoting $maxmin(x_1, .., x_n) = max(x_1, .., x_n) - min(x_1, .., x_n)$ we have

$$\begin{aligned}
F^0(x^0) &= -5x^0_1 + x^0_2 - x^0_3 + maxmin(1, x^0_1, x^0_2) - 1 + maxmin(x^0_1, x^0_2, x^0_3) \\
F^1(x^1) &= -3x^1_1 + 3x^1_2 + x^1_3 + maxmin(1, x^1_1, x^1_2) - 1 + maxmin(x^1_1, x^1_2, x^1_3) \\
F^2(x^2) &= -x^2_1 + 5x^2_2 + 3x^2_3 + maxmin(1, x^2_1, x^2_2) - 1 + maxmin(x^2_1, x^2_2, x^2_3)
\end{aligned}$$

It is simple to check that, the minimizers of each binary MRF energy is given as follows

$$x^{\star 0} = (1, 1, 1)$$

$$x^{\star 1} = (1, 0, 0)$$

$$x^{\star 2} = (0, 0, 0)\,,\; x^{\star 2} = (1, 0, 0)$$

Note that there exist two minimizers for F^2 and that this scheme gives two minimizers of the energy F

$$x^\star = (2,1,1)\,,\; x^\star = (3,1,1)$$

both of which give the minimum energy -8. $\square$

The algorithm we propose is not the one we described above. Instead we employ an obvious improvement. For the first layer we pick $i = L/2-1$. Note that the minimizer $x^{\star L/2-1}$ of $F^{L/2-1}$ provides the most significant bits of a minimizer of F. We then construct another MRF with a narrower intensity space $\Lambda = \{0,..,L/2-1\}$ as follows

$$G(x) = \sum_{k=1}^{K}\sum_{\pi\in\Pi^k} g_\pi(x_{\pi_1},..,x_{\pi_k})$$

where for any clique $\pi \in \Pi^k$

$$g_\pi(x_{\pi_1},..,x_{\pi_k}) = f_\pi(2^N x^{\star L/2-1}_{\pi_1} + x_{\pi_1},..,2^N x^{\star L/2-1}_{\pi_k} + x_{\pi_k})$$

where $N = log(L)-1$. We proceed to find the binary minimizer of $G^{L/4-1}$, hence the second most significant bits of the minimizer of F and so on. More formally the following is the algorithm we propose.

1. Assign $N = log(L)-1$. Set $x^\star_s = 0$ for $s\in\Sigma$.

2. If $L = 1$ terminate, the image $x^\star$ is the minimizer. Otherwise set

$$F(x) = \sum_{k=1}^{K}\sum_{\pi\in\Pi^k} f_\pi(x^\star_{\pi_1} + x_{\pi_1},..,x^\star_{\pi_1} + x_{\pi_k})$$

   Calculate the minimizer $y^\star$ of $F^{L/2-1}$.

3. Reset the image $x^\star$ as follows

$$x^\star_s = x^\star_s + 2^N y^\star_s$$

   for any $s\in\Sigma$. Reset $L = L/2$, $N = N-1$ and go to step 2.

Note that, this algorithm holds only if all of the shifted potentials f_π of the modified MRF energy of step 2 given as

$$F(x) = \sum_{k=1}^{K} \sum_{\pi \in \Pi^k} f_\pi(x^\star_{\pi_1} + x_{\pi_1}, .., x^\star_{\pi_1} + x_{\pi_k})$$

are levelable.

Example 3.35 We demonstrate the algorithm with the same MRF energy as we gave in example 3.34. Recall that up to a constant

$$F(x) = (x_1 - 3)^2 + x_2^2 + (x_3 - 1)^2 + maxmin(3, x_1, x_2) + maxmin(x_1, x_2, x_3)$$

we initialize $x^\star_1 = x^\star_2 = x^\star_3 = 0$ and $N = log(L) - 1 = 1$. We already found the minimizer $y^\star$ of $F^{L/2-1} = F^1$ in example 3.34 as $y^\star = (1, 0, 0)$. So we set $x^\star_s = 2^N y^\star_s$, for $s = 1, 2, 3$. Thus $x^\star = (2, 0, 0)$.

We reassign $L = 2$, $N = 0$ and construct

$$\begin{aligned} F(x) &= (x^\star_1 + x_1 - 3)^2 + (x^\star_2 + x_2)^2 + (x^\star_3 + x_3 - 1)^2 \\ &\quad + maxmin(3, x^\star_1 + x_1, x^\star_2 + x_2) \\ &\quad + maxmin(x^\star_1 + x_1, x^\star_2 + x_2, x^\star_3 + x_3) \\ &= (2 + x_1 - 3)^2 + x_2^2 + (x_3 - 1)^2 + maxmin(3, 2 + x_1, x_2) \\ &\quad + maxmin(2 + x_1, x_2, x_3) \end{aligned}$$

Now we need to check the levelability of each potential above. The first order potentials are obviously levelable. Note that since $x_1, x_2, x_3 \in \Lambda_L = \Lambda_2$, for the second and the third order potentials we have

$$maxmin(3, 2 + x_1, x_2) = 3 - x_2$$
$$maxmin(2 + x_1, x_2, x_3) = 2 + x_1 - min(x_2, x_3)$$

Thus we have, up to a constant

$$F(x) = (2 + x_1 - 3)^2 + x_1 + x_2^2 - x_2 + (x_3 - 1)^2 - min(x_2, x_3)$$

which is a levelable MRF energy of order two. Note that, $F^{L/2-1} = F^0 = F$ is minimized by two binary images $(0, 1, 1)$ and $(1, 1, 1)$. We can pick any of them. If we pick $y^\star = (0,1,1)$, we recalculate $x^\star_s = x^\star_s + 2^N y^\star_s = x^\star_s + y^\star_s$ for $s = 1,2,3$. Thus we have $x^\star = (2,1,1)$. If we pick $y^\star = (1,1,1)$ on the other hand, we recalculate $x^\star_s = x^\star_s + y^\star_s$ for $s = 1,2,3$ to get $x^\star = (3,1,1)$. We reassign $L = L/2 = 1$ and terminate at the step 2. Thus the minimizers of the initial MRF energy are found as $(3,1,1)$ and $(2,1,1)$ as expected. In applications, we do not aim to find all minimizers of an MRF, we just pick any one of them. $\square$

Until now we have not addressed how to minimize binary MRFs. For this issue we refer to [22] where it is proved that the problem of minimization of a regular binary MRF of order up to three is equivalent to a maximum flow computation [1, 3] on an appropriately defined graph. For the details, including how the graph is constructed, we refer to [22].

In [10, 18, 8, 33] similar algorithms for second order MRF energies are presented. For a comparison, in those references basically it is assumed that if

$$
\begin{aligned}
&i. \quad f_\pi \text{ is convex if } \pi \in \Pi^1 \\
&ii. \quad f_\pi \text{ is given by } f_\pi(x_{\pi_1}, x_{\pi_2}) = c_\pi |x_{\pi_1} - x_{\pi_2}| \text{ if } \pi \in \Pi^2
\end{aligned}
\tag{3.26}
$$

where $c_\pi \in \mathbb{R}$ is nonnegative for any $\pi \in \Pi^2$, the MRF energy F given by

$$F(x) = \sum_{\pi \in \Pi^1} f_\pi(x_{\pi_1}) + \sum_{\pi \in \Pi^2} f_\pi(x_{\pi_1}, x_{\pi_2})$$

can be minimized by similar algorithms. Note that for any $\pi \in \Pi^2$ the second order potentials defined by (3.26) are levelable. Moreover the potentials given by (3.26) are valuable for image processing, as they are used to discretize the total variation prior [10] which is widely used in various fields of image processing [9, 2]. Hence we contributed to the subject by showing that the algorithm can be used to minimize MRF energies with an extended number of priors, including higher order priors. In the second part [11] of the paper [10], a definition for levelable functions without the conditions H1 and H2 is given. However the algorithm given for a class of MRF energies with levelable functions is not the one given in [10] and is similar to the one given in [19] and requires larger graphs.

We need to point out that the algorithm we proposed is slightly different than the one given in [10] and almost the same as the one given in [8]. It has an additional advantage that one can terminate the iterations early to find a predefined number of most significant bits of a minimizer. This scheme can be used for faster, inexact minimization.

In [18] another improvement to our minimization scheme is provided. There, the author proposed a minimization scheme, to minimize $log(L)$ binary MRFs, with complexity of just one binary MRF minimization. An implementation of the proposed scheme is given in [16] with comparisons with the algorithm given in [10]. We did not implement that scheme.

3.2.5 Numerical Results

In this section we are going to give some applications of the results of the previous sections to image denoising. As usual we set $L = 256$ and $N = log(L) = 8$, hence our method basically is composed of 8 maximum flow computations. As we previously mentioned, the scope of this work is not aimed to cover either the theory or the implementation of maximum flow problems. For the details of those, we refer to the texts [1, 3], which we find more comprehensible among various similar books on the subject.

We also did not implement any maximum flow algorithms, we instead used the BOOST Graph Library [30], BGL hereinafter, for this purpose. We need to mention that, although one would expect that this is a well written library, it is not intended to be used in image processing. Therefore it is not optimized howsoever as an image processing library hence can not be considered to be the ultimate tool for our purposes. As we shall soon see however, the implementations given in the BGL help us to compare the available maximum flow algorithms and they are likely to be starting points for faster implementations.

To be able to use the BGL one needs an appropriate interface to the library. We encountered such an application in MATLAB File Exchange web site, the MatlabBGL library [15] provided by David Gleich. We used MatlabBGL to interface the graphs we constructed to the BGL. As we previously mentioned, we used the perscription given in [22] for graph construction. We wrote a C++

library which we built as a mex file[12]. This way we were able to use MATLAB essentially as a user interface tool which can handle jobs which are not time critical. We used Microsoft Visual Studio 2005 as the C++ compiler and MATLAB R2007b on a Windows XP PC with Intel Pentium 4 at 2800 MHz and 1GB memory. We also employed built-in profile guided optimization tool of Visual Studio 2005.

We are going to demonstrate the results using two different topologies; one with first and second order cliques only and another one with first and third order cliques only, see Figure 3.3.

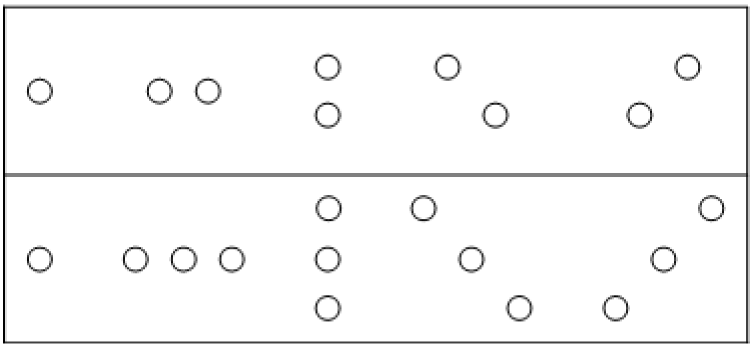

Figure 3.3: The cliques of the topologies used for the examples.

In the sequel we assume that any clique mentioned is one of the cliques sketched in Figure 3.3. For any second or third order clique π, we are going to use the potential function

$$f_\pi(x_{\pi_1},..,x_{\pi_n}) = a \cdot (max(x_{\pi_1},..,x_{\pi_n}) - min(x_{\pi_1},..,x_{\pi_n}))$$

Here of course n is either 2 or 3 and a is a nonnegative integer. Notice that when $n = 2$, the equation above becomes

$$f_\pi(x_{\pi_1}, x_{\pi_2}) = a \cdot |x_{\pi_1} - x_{\pi_2}|$$

We denote the noisy image by y and we denote the estimatee by x. We performed our tests on the test images given in Figure 3.4. All images are of size 256x256. In the Figure 3.5 the noisy versions of the images are given. The noise

[12] The libraries which are built as mex files are accessible by MATLAB

is white Gaussian with zero mean and standard deviation 10 and is calculated by MATLAB's appropriate tools.

We need to recall here that our method is composed of 8 steps. We initialize every pixel of the image x with zero intensities and at the first step we recover the most significant bits of intensities of each pixel, at the second step we recover the second most significant bits of intensities of each pixel and so forth. Therefore for the sake of timing one can terminate the iterations at an earlier step than the 8th, to obtain an image intensities of each pixels of which are composed of the first few bits of those of the minimizer image. This is a nonexact approach of course.

We are going to deal with the following MRF energies

$$F(x) = \sum_{\pi \in \Pi^1} (x_{\pi_1} - y_{\pi_1})^2 + \sum_{\pi \in \Pi^2} a_1 |x_{\pi_1} - x_{\pi_2}| \tag{3.27}$$

$$G(x) = \sum_{\pi \in \Pi^1} (x_{\pi_1} - y_{\pi_1})^2 + \sum_{\pi \in \Pi^3} a_2 (max(x_{\pi_1}, x_{\pi_2}, x_{\pi_3}) - min(x_{\pi_1}, x_{\pi_2}, x_{\pi_3})) \tag{3.28}$$

with $a_1 = 11$ and $a_2 = 6$. We have to mention a point here. Recall that the coefficients a_1 and a_2 determine the capacities of the arcs when we construct the appropriate graphs for maximum flow computations and recall that we would like to have integer capacities for arcs. This is the general practice for maximum flow computations although theoretically they apply for graphs with arcs of non-integer capacities. We are going to stick with the common practice and deal with integer coefficients for numerical examples.

The images given in Figure 3.6 are the minimizers of F given by (3.27). We plot the associated residual images in Figure 3.7 which are intentionally biased for easy visual perception. Similarly the minimizers of G given by (3.28) and the associated residual images are plotted in the Figures 3.8 and 3.9 respectively. Note the excellent restoration performances of both vertical and horizontal edges especially on the Checker Board image. In general the restored edges do not suffer from blurring. However notice also that the texture parts of the natural images are substantially smoothed out. Note also the sketchy character of the smooth

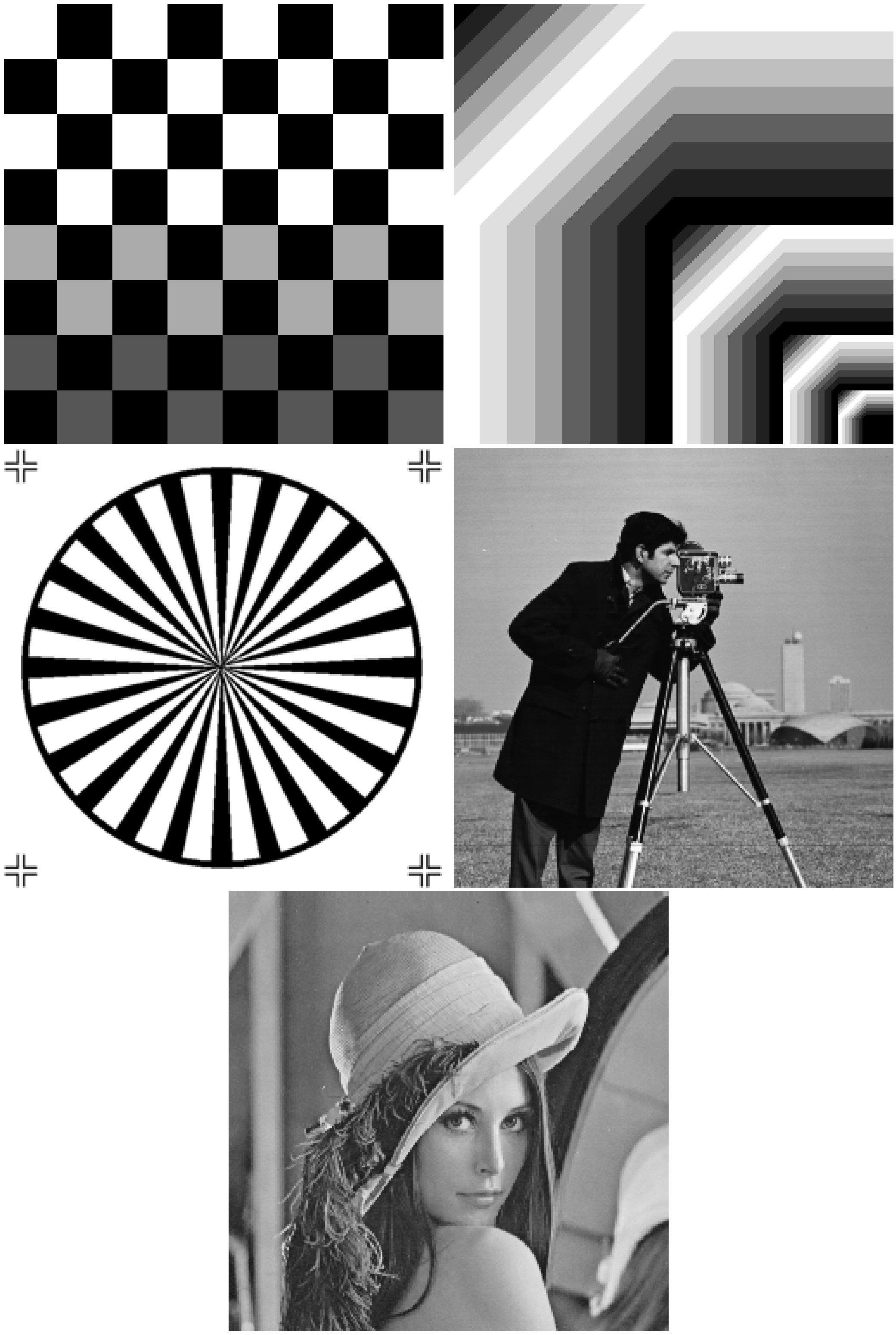

Figure 3.4: Test images. The Checker Board on the first row left, the Pipes on the first row right, the Test Pattern on the second row left, the Cameraman on the second row right and Lena at the bottom.

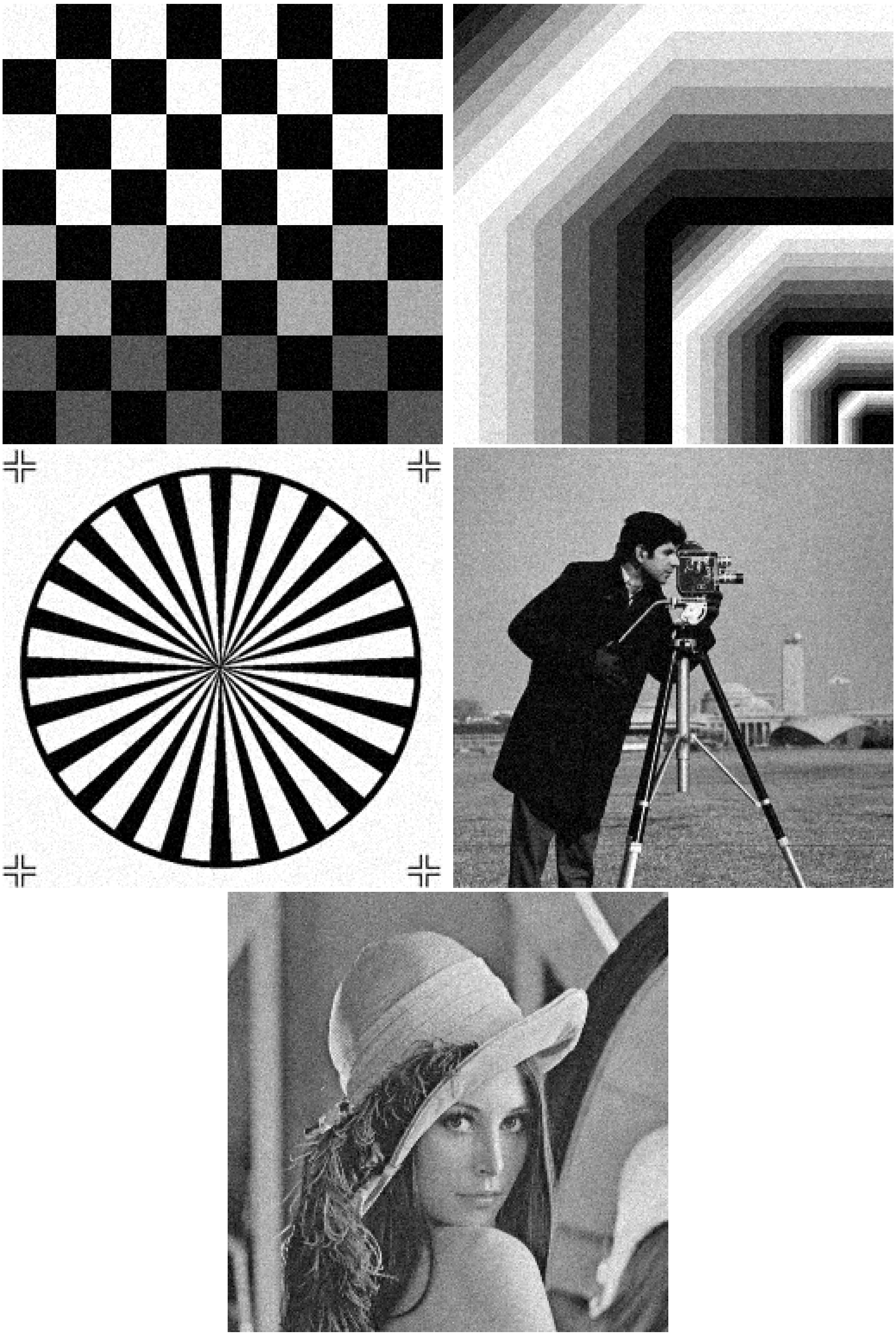

Figure 3.5: Images contaminated by white Gaussian noise of zero mean and 10 standard deviation.

sections of both minimizers. The restoration performances of minimizers of G are slightly superior to those of minimizers of F.

The images given in Figure 3.10 and 3.12 are the results of early termination, i.e., the most significant 6 bits of the minimizers of F and G respectively. The associated residual images are plotted in Figure 3.11 and Figure 3.13 respectively. Note that it is hardly possible to discriminate these images among the corresponding minimizer images.

There exist three maximum flow algorithms in the BGL. Two of them are Edmunds-Karp and push-relabel algorithms [1, 3, 30]. The last algorithm by Kolmogorov [5], is a newer one. In fact the current 1.34 release of the BGL does not contain an implementation of it, however it is available in the Internet[13]. The performance of Edmunds-Karp implementation in the BGL was so slow that we do not give its evaluations here. In Table 3.1 and Table 3.2, we tabulated the PSNR, energy and timing figures related to the minimization of the energy function F given in (3.27) and the energy function G given in (3.28) respectively. The PSNR figures are calculated according to the following formula.

$$PSNR(x) = 10log(\frac{255^2}{(\frac{1}{M}\sum_{i\in\Sigma}(x_i - z_i)^2)})$$

where z denotes the corresponding original image and M is the number of pixels in Σ. One thing we immediatiely notice is that, the Kolmogorov algorithm clearly outperforms push-relabel algorithm. This is interesting, since the Kolmogorov algorithm is merely a variant of Edmunds-Karp algorithm which has a greater complexity than that of the push-relabel algorithm [5]. For both algorithms timing figures heavily depend on the test images and the coefficients. This is reasonable since the execution times of maximum flow algorithms typically depend on the dynamic range of the capacities of the arcs in the graphs. We need to stress that minimization of F is faster than minimization of G when $a_1 = a_2$. This is expected since there are more arcs in the graph associated to G. However when $a_2 < a_1$ timing performances of two minimization tasks may

[13] The MatlabBGL library has an implementation. A few other places may be found upon a Google Code Search.

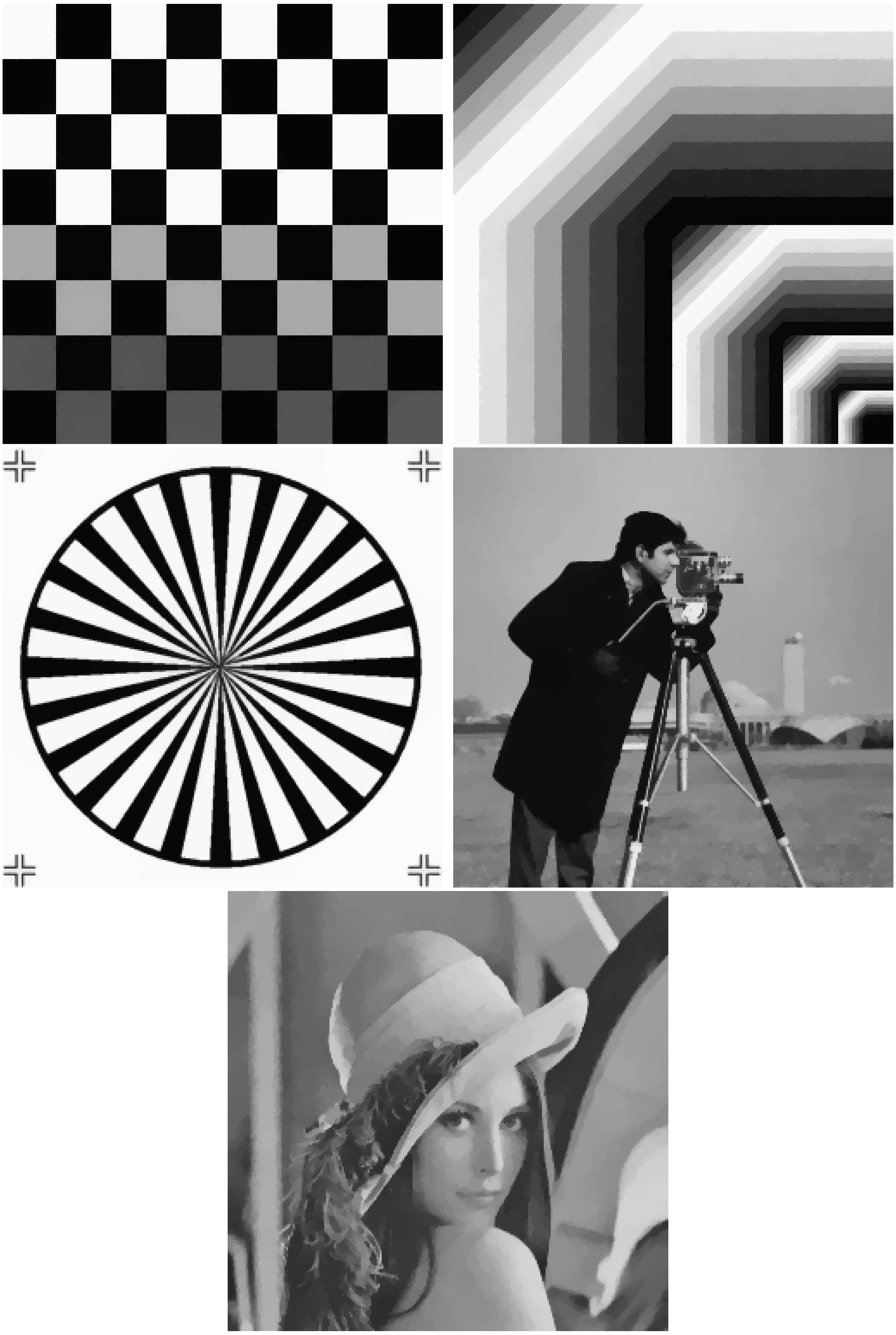

Figure 3.6: The minimizers of F given in Equation (3.27).

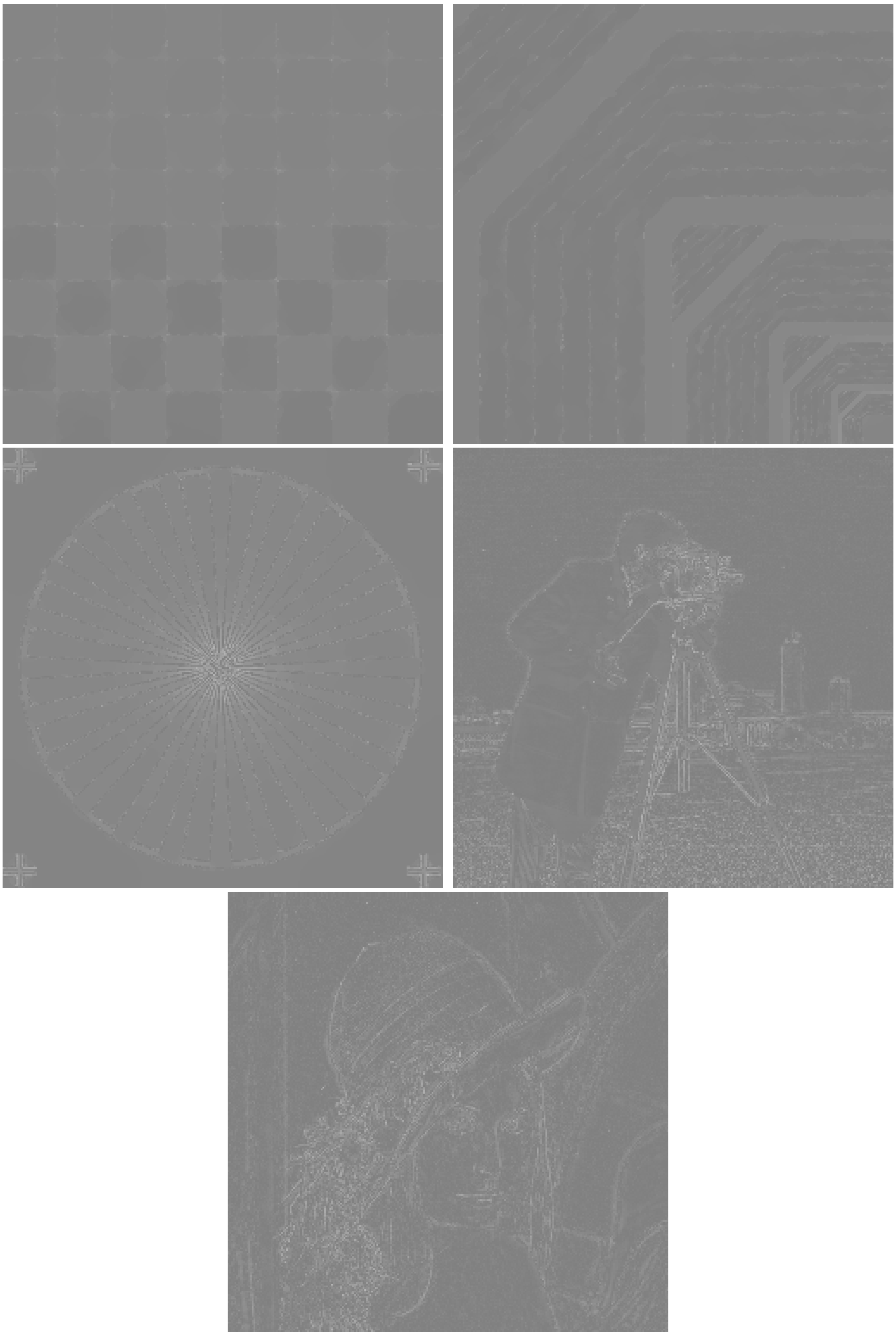

Figure 3.7: The residual images associated with the minimizers of the energy F given in Equation (3.27).

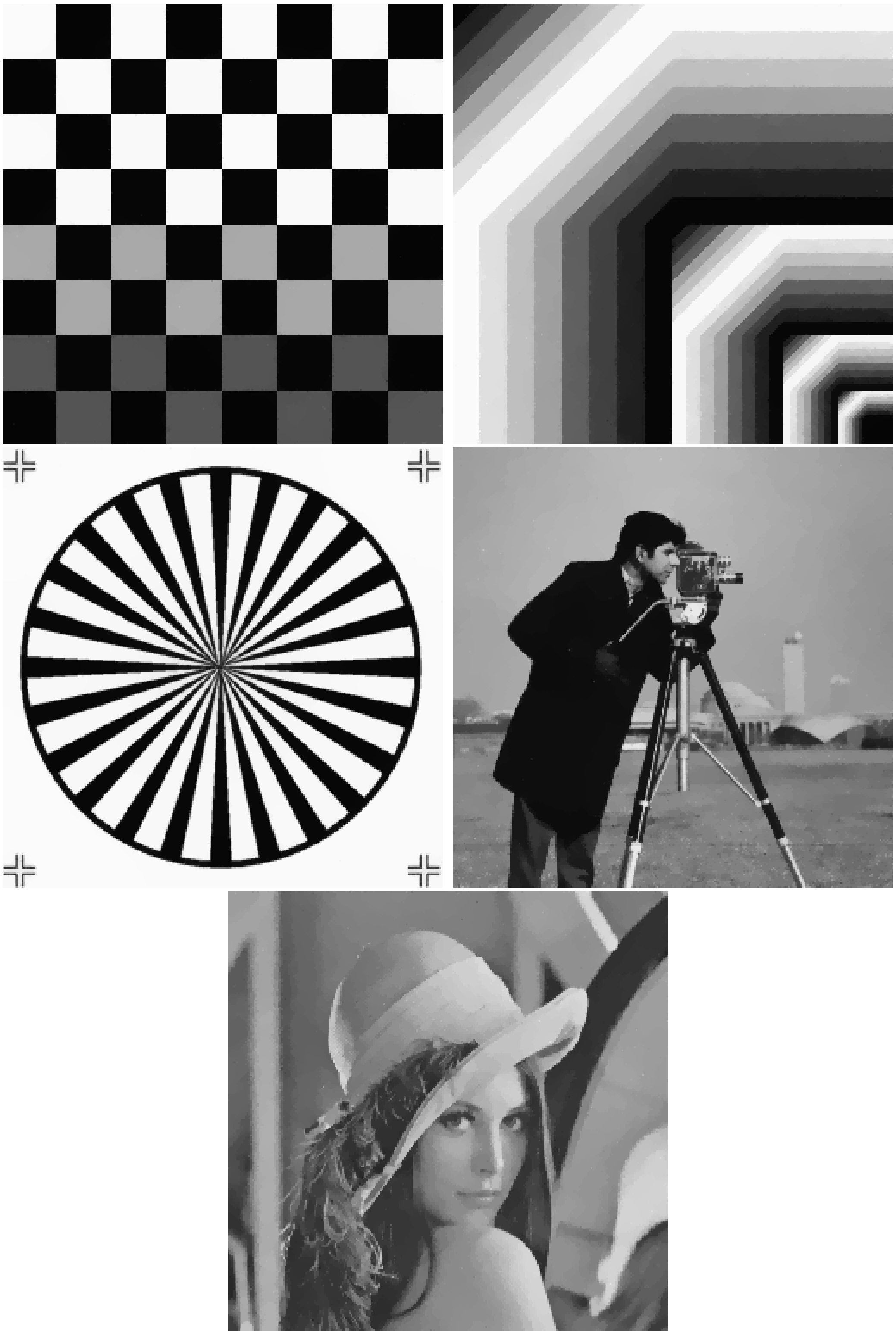

Figure 3.8: The minimizers of G given in Equation (3.28).

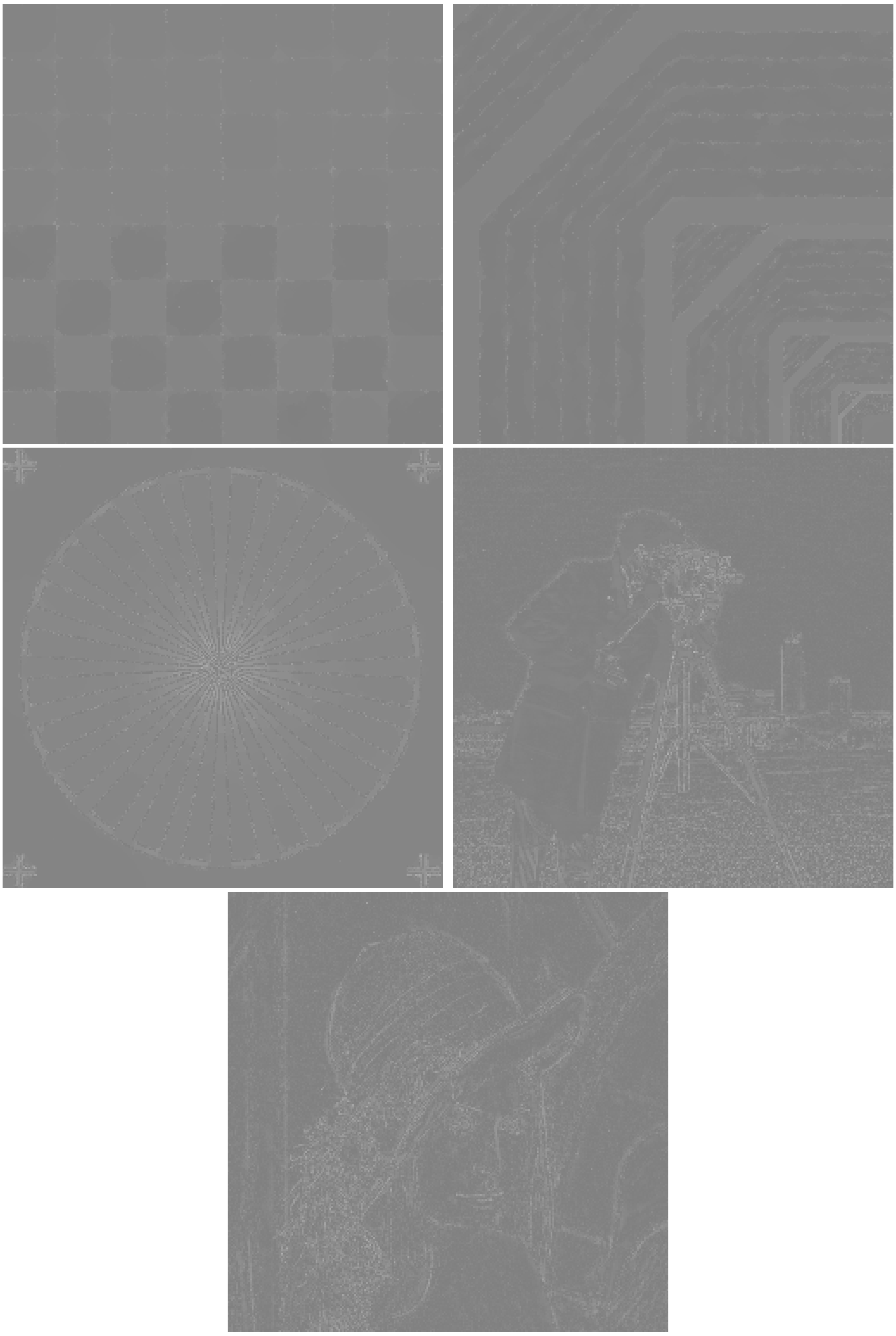

Figure 3.9: The residual images associated with the minimizers of the energy G given in Equation (3.28).

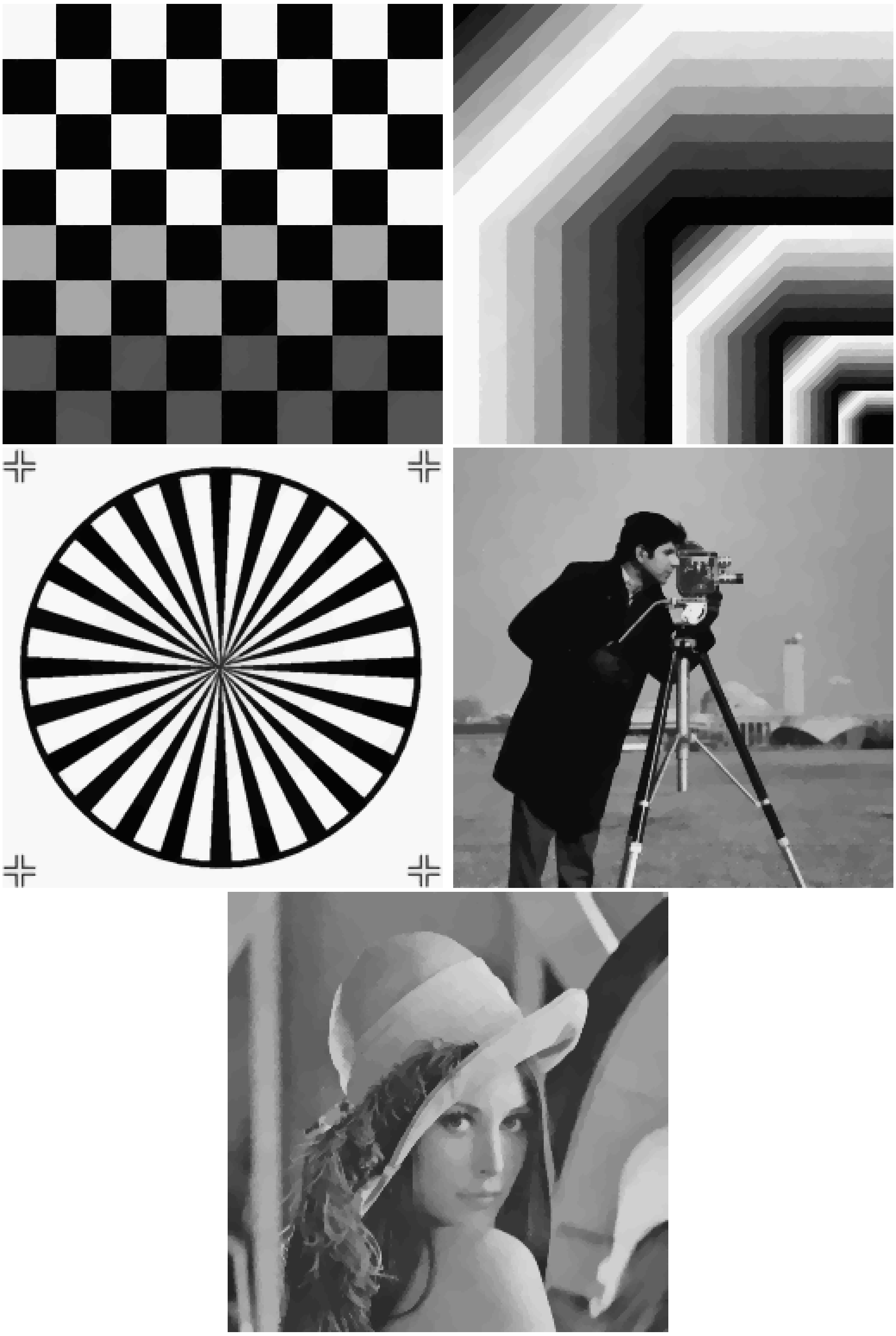

Figure 3.10: The most significant 6 bits of the minimizers of the energy function F given in Equation (3.27).

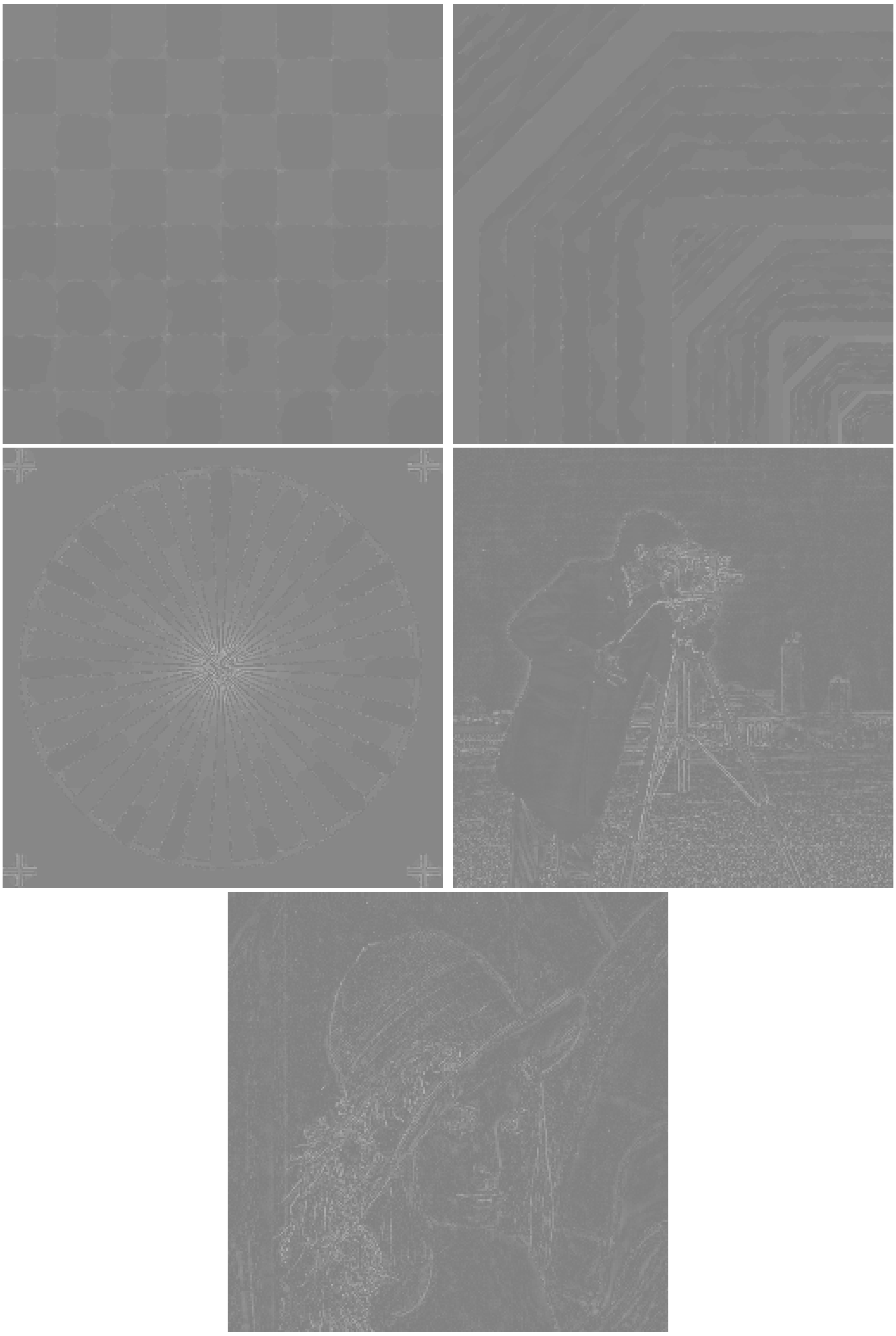

Figure 3.11: The residual images associated with the most significant 6 bits of the minimizers of the energy F given by Equation (3.27).

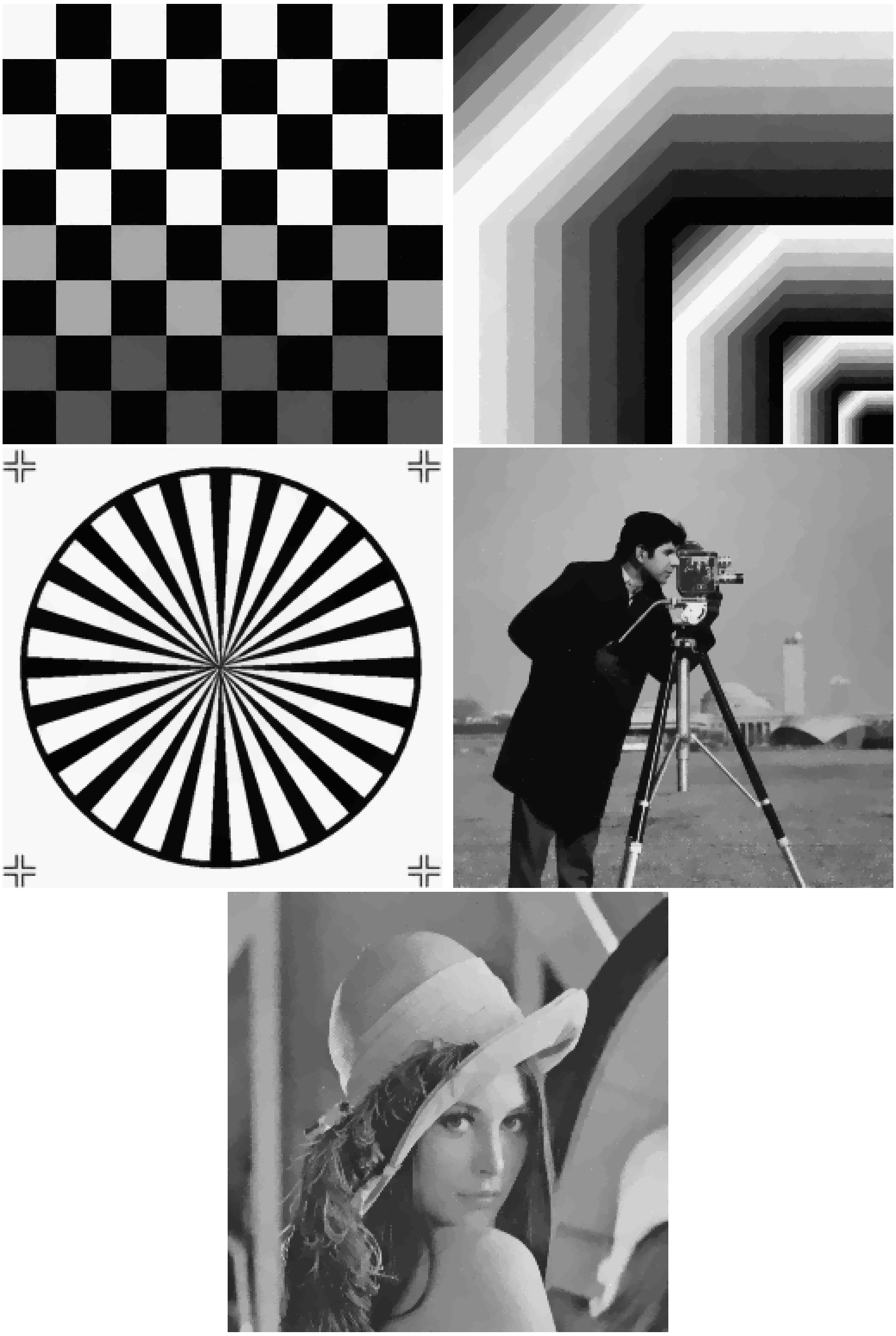

Figure 3.12: The most significant 6 bits of the minimizers of the energy function G given in Equation (3.28).

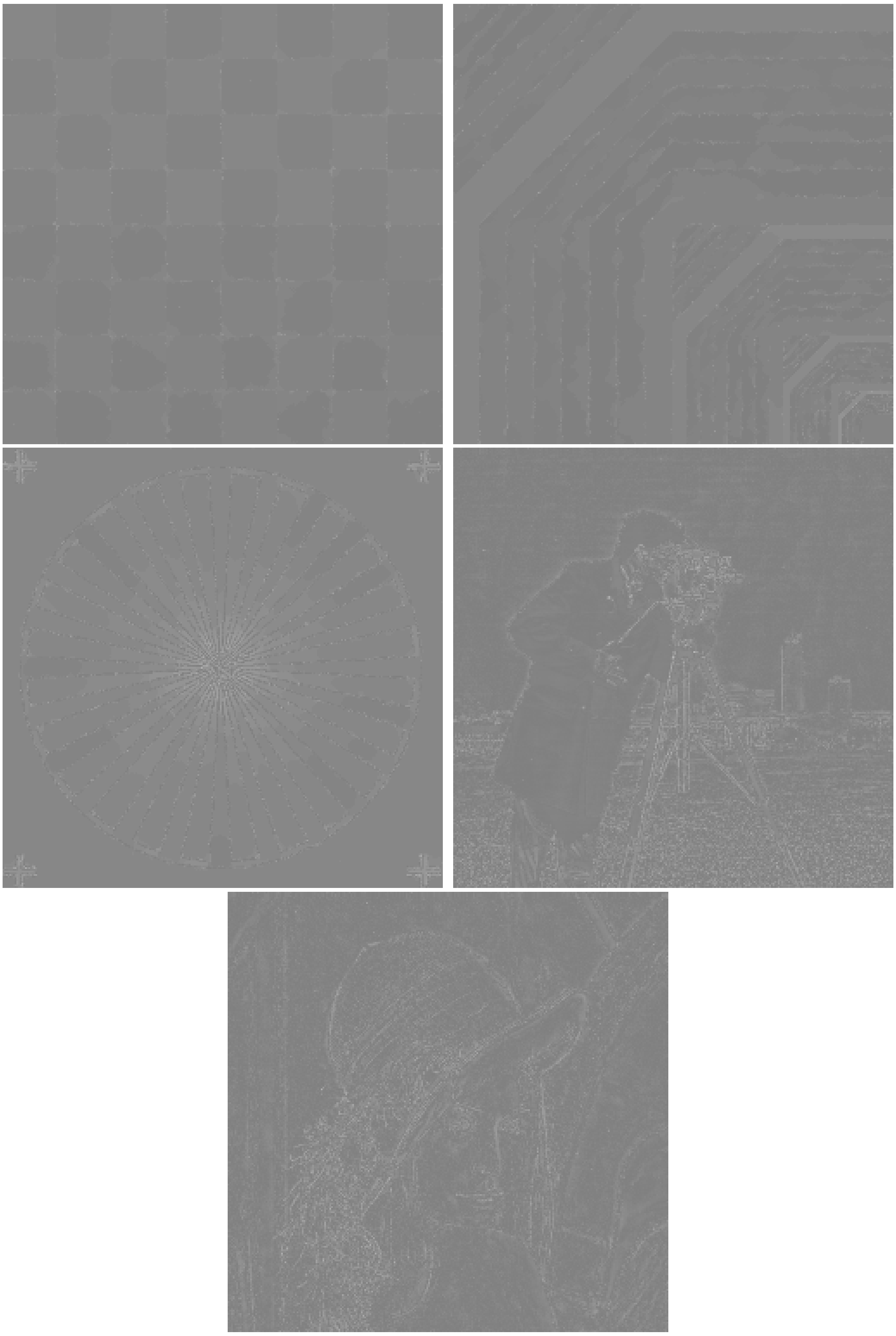

Figure 3.13: The residual images associated with the most significant 6 bits of the minimizers of the energy G given by Equation (3.28).

become comparable.

Another interesting point is that, the algorithm barely needs to be run to the end. The most significant 6 or 7 bits always give accurate estimates according to the energy, PSNR figures and subjective opinion. According to the PSNR figures minimizers of F are superior to the minimizers of G. This is not in accordance with our observations. However we can not claim that energy function G is substantially superior to F in terms of the restoration performances either. Their performances are similar and minimizers of G are slightly more pleasant than those of F. We found that the PSNR figures are not always reliable, for example observe that for Test Pattern image, the PSNR figures of minimizers of both energies are smaller than the PSNR of the noisy image.

Finally we would like to present a comparison of the denosing performance of our methods with that of a simple averaging and another denoising algorithm. For averaging we use a 3x3 Gaussian kernel with standard deviation 0.5 generated by suitable MATLAB tools. We also compare our results with the denoising results of the anisotropic LPA-ICI recursive denoising algorithm proposed in the paper [12]. We did not implement this algorithm. Instead we use the performance figures given in the cited web site. We tabulated PSNR figures of the denoising methods in 3.3. We used the Cameraman image for this experiment. We picked PSNR wise best set of coefficients a_1 and a_2 for energy functions F and G respectively. This table shows that minimizers of both energy functions are vastly superior to simple averaging according to PSNR figures. This is also justified by our subjective decision. Moreover notice that according to PSNR figures the performance of LPA-ICI is comparable to ours. We need to mention that employing adaptation schemes on the coefficients a_1 and a_2 could enhance the denoising performance of our methods. We do not have the timing figures of the LPA-ICI method so we can not provide the corresponding comparison.

As a result we can claim that the denoising scheme via minimization of MRF energies F and G bring a certain quality to the denosing results. Observe that MRF energy minimization scheme is inherently space variant, i.e., one can tune

		Noisy Image	6	7	8
Checker Box	PSNR	30.11	33.98	33.54	33.65
	Energy Rate	01.67	01.01	01.00	01.00
	Time-PR	n/a	07.25	11.42	15.05
	Time-KOL	n/a	01.20	02.00	02.66
Pipes	PSNR	28.67	34.97	36.20	36.37
	Energy rate	02.49	01.02	01.00	01.00
	Time-PR	n/a	07.79	10.27	14.15
	Time-KOL	n/a	01.37	01.81	02.38
Test Pattern	PSNR	30.60	28.55	29.01	29.05
	Energy rate	01.25	01.00	01.00	01.00
	Time-PR	n/a	06.89	09.47	15.52
	Time-KOL	n/a	01.31	01.75	02.59
Cameraman	PSNR	28.27	29.18	29.32	29.35
	Energy rate	02.14	01.01	01.00	01.00
	Time-PR	n/a	07.54	10.43	13.59
	Time-KOL	n/a	01.47	02.01	02.57
Lena	PSNR	28.13	29.37	29.56	29.59
	Energy rate	02.10	01.01	01.00	01.00
	Time-PR	n/a	09.44	11.89	14.70
	Time-KOL	n/a	01.57	02.00	02.43

Table 3.1: The PSNR, energy and timing figures for the minimization of the energy function F given by Equation (3.27). The last three columns represent the images constructed using the most significant 6, 7 and 8 bits of the intensities of the associated minimizer. Energy rate is the energy of the corresponding image divided by the energy of the minimizer. Time-PR is the total time necessary for calculation of the images if push-relabel algorithm is used for maximum flow computation. Time-KOL is the corresponding time if Kolmogorov algorithm is used. Timing figures are in seconds. The noise is white Gaussian with zero mean and 10 standard deviation.

		Noisy Image	6	7	8
Checker Box	PSNR	30.17	34.04	33.52	33.61
	Energy Rate	01.48	01.01	01.00	01.00
	Time-PR	n/a	06.89	10.33	13.16
	Time-KOL	n/a	01.77	02.94	03.94
Pipes	PSNR	28.68	34.88	36.01	36.13
	Energy rate	02.05	01.02	01.00	01.00
	Time-PR	n/a	07.45	10.12	12.83
	Time-KOL	n/a	01.93	02.67	03.46
Test Pattern	PSNR	30.65	28.35	28.93	28.99
	Energy rate	01.18	01.00	01.00	01.00
	Time-PR	n/a	06.86	09.34	13.06
	Time-KOL	n/a	01.78	02.40	03.47
Cameraman	PSNR	28.28	29.61	29.78	29.81
	Energy rate	01.83	01.01	01.00	01.00
	Time-PR	n/a	07.56	10.44	13.25
	Time-KOL	n/a	02.67	03.69	04.71
Lena	PSNR	28.11	29.63	29.83	29.86
	Energy rate	01.79	01.01	01.00	01.00
	Time-PR	n/a	09.21	11.58	14.09
	Time-KOL	n/a	02.92	03.69	04.46

Table 3.2: The PSNR, energy and timing figures for the minimization of the energy function G given by Equation (3.28). The last three columns represent the images constructed using the most significant 6, 7 and 8 bits of the intensities of the associated minimizer. Energy rate is the energy of the corresponding image divided by the energy of the minimizer. Time-PR is the total time necessary for calculation of the images if push-relabel algorithm is used for maximum flow computation. Time-KOL is the corresponding time if Kolmogorov algorithm is used. Timing figures are in seconds. The noise is white Gaussian with zero mean and 10 standard deviation.

Standard Deviation	PSNR				
	Noisy Image	Averaging	Minimizer of F	Minimizer of G	LPA-ICI
5	34.15	32.54	36.84	36.78	37.74
10	28.12	29.93	32.62	32.32	33.35
15	24.61	27.54	30.17	29.85	31.09
20	22.11	25.55	28.61	28.22	29.74
25	20.13	23.85	27.39	26.92	28.68
30	18.62	22.47	26.37	25.92	27.76
35	17.24	21.22	25.46	25.04	26.94
50	14.17	18.44	23.43	23.03	24.92

Table 3.3: PSNR figures of various denoising methods. The first column tabulates the standard deviation of the white Gaussian noise used in each experiment.

the coefficients a_1 or a_2 in a pixel dependent manner accroding to the spatial features of the image in consideration. This would presumably enhance the performance of the method.

CHAPTER 4

CONCLUSION AND FUTURE WORK

We propose an efficient and exact energy minimization algorithm which is based on the novel graph-cuts techniques. The main contribution of our work is the introduction of the use of higher order cliques into image restoration using second class of algorithms[1]. We note that for the third class of methods there has been an attempt in [21] for higher order cliques. We take an abstract approach to the problem and provide a fairly detailed investigation of binary MRFs. We think that the definitions and the notation we introduced for the study of binary MRFs may be of further use, perhaps for other fields too. Moreover we give a property of minimizers of regular binary MRFs, namely Theorem 3.15 which we think may be further utilized in its own.

We also give a generalization of the monotonicity property of minimizers of binary MRFs [10, 8, 18, 33] which has been known for several years. This property may also be helpful for construction of minimization algorithms for new, more accurate energy functions.

Graph-cuts based techniques are still evolving. As opposed to their novelty in the image processing, they have attracted great attention in almost all fields in image processing. This is partly because of the efficiency and accuracy they provide and partly due to the generality of the energy minimization framework. However currently it is not still possible to build real time applications using them.

[1] Please see the introductory section of Chapter 3 for the second and third classes of graph-cuts based methods.

The main reason for this is the bulding blocks of the graph-cuts based methods, maximum flow computations. As we mentioned before, maximum flow problem is one of the most studied problems in network flow theory and integer programming. However, either because of the graph sizes encountered in those areas are smaller or because of the lack of demand to real time algorithms, apperently not enough effort has been made to speed up the existing algorithms[2].

The main difficulty in implementing maximum flow algorithms is the inherent misuse of the memory. To update any information of a node or edge, a typical maximum flow algorithm requires the information from virtually any node or edge of the graph. This requires frequent caching and uncaching of memory which degrades the performance of the implementation. This may be unavoidable too, however for fast image processing algorithms, especially for embedded applications for which memory resources are scarce, making good use of the cache is vital. This is the first point we need to mention for future directions. The need for cache friendly algorithms is also pointed out by the authors in [5], where they offered preflow based algorithms [3] for cache directed improvements[3]. However they gave an improvement to the augmenting path based algorithms [1] and gathered a substantial uplift. We note that they did not offer any cache aware implementation either.

Apart from cache directed improvements, there is one more way of speed improvement which is proposed in [18] and implemented in [16]. In [18] a similar method to ours, which hold for second order cliques only, is proposed. Further, an improvement of that method is offered which could accomplish the work done by 8 consecutive maximum flow computations at once in single maximum flow computation complexity. The implementation in [16] proved the improvement though in only moderate levels. That kind of an implementation has not been done yet for higher order cliques.

An obvious drawback of our algorithm is the highly small number of priors

[2] This statement is up to our knowledge of course.
[3] The push relabel algorithm of the BGL is a preflow based algorithm.

we can use, highly limited number of levelable functions there exist. The reason we have very few number of levelable functions is our definition of levelability, namely Equation (3.24). It is indeed possible to decompose any function into functions of binary levels of variables [34][4]. However this decomposition is coupled, i.e., each function in the decomposition is in terms of mixed levels of the variables. We refer the reader to [34] for the details. To be able to use this decomposition efficiently, one requires a more powerful result than monotonicity.

[4] Exactly the same way we decomposed functions of single variables.

APPENDIX A

Regularity of Functions

Definition A.1 *For any $f \in V_2^n$, we say f is **two-regular** if all $(p; q_1, q_2)$ regularities of f are nonnegative when $0 \le q_1, q_2 \le 1$.*

The following is obvious.

Theorem A.2 *Any projection of a two-regular function is two-regular.*

In [22], authors defined regular functions as the ones which are two-regular. It is obvious that our definition of regularity implies two-regularity. The following proves that the converse is also true.

Theorem A.3 *Any two-regular function $f \in V_2^n$ is regular.*

Proof If $n = 1$ or $n = 2$ two-regularity of $f \in V_2^n$ obviously implies regularity of f. We start an induction; we set a positive integer $n > 2$ and assume that two-regularity of $f \in V_2^{\bar{n}-1}$ implies regularity of f for $\bar{n} \le n$. Let $f \in V_2^n$ be two-regular, we shall prove that f is regular.

Let the collection $\{\alpha^0; \alpha^1, \alpha^2\}$ be a $(p; q_1, q_2)$ partition of I_n and assume that $a = (a_1, .., a_p) \in \Lambda_2^p$ is given. If $0 \le q_1, q_2 \le 1$, there is nothing to prove so assume $q_1 \ge 2$. Let

$$\bar{\alpha}^0 = \{\alpha_1^0, .., \alpha_p^0, \alpha_1^1\}$$
$$\bar{\alpha}^1 = \alpha^1 \setminus \{\alpha_1^1\}$$

define $\bar{a} \in \Lambda_2^{p+1}$ as follows

$$\bar{a} = (\bar{a}_1, .., \bar{a}_{p+1}) = (a_1, .., a_p, 1)$$

and define $\bar{f} \in V_2^{n-1}$ as

$$\bar{f} = \mathcal{P}^{(\bar{a}_1..\bar{a}_{p+1})}_{\bar{\alpha}^0;\bar{\alpha}^1,\alpha^2}(f)$$

Due to Theorem A.2, $\bar{f}$ is two-regular, therefore regular due to the induction hypothesis. Thus

$$\bar{f}(1,0) + \bar{f}(0,1) - \bar{f}(0,0) - \bar{f}(1,1) \geq 0 \tag{A.1}$$

Now, let

$$\begin{aligned}\tilde{\alpha}^0 &= \{\alpha_1^0, .., \alpha_p^0, \alpha_2^1, .., \alpha_{q_1}^1\}\\ \tilde{\alpha}^1 &= \{\alpha_1^1\}\end{aligned}$$

define $\tilde{a} \in \Lambda_2^{p+q_1-1}$ as

$$\tilde{a} = (\tilde{a}_1, .., \tilde{a}_{p+q_1-1}) = (a_1, .., a_p, \overbrace{0, .., 0}^{q_1 - 1 \text{ times}})$$

and define $\tilde{f} \in V_2^{n-q_1+1}$ as follows

$$\tilde{f} = \mathcal{P}^{(\tilde{a}_1..\tilde{a}_{p+q_1-1})}_{\tilde{\alpha}^0;\tilde{\alpha}^1,\alpha^2}(f)$$

Again, due to Theorem A.2, $\tilde{f}$ is two-regular, therefore regular due to the induction hypothesis. Thus

$$\tilde{f}(1,0) + \tilde{f}(0,1) - \tilde{f}(0,0) - \tilde{f}(1,1) \geq 0 \tag{A.2}$$

Notice that for any $b \in \Lambda_2$

$$\bar{f}(0,b) = \tilde{f}(1,b)$$

Define

$$\hat{f} = \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1,\alpha^2}(f)$$

and notice due to definitions of $\bar{f}$ and $\tilde{f}$ that

$$\begin{aligned}\hat{f}(1,b) &= \bar{f}(1,b)\\ \hat{f}(0,b) &= \tilde{f}(0,b)\end{aligned}$$

Therefore adding up equations (A.1) and (A.2) gives

$$\hat{f}(1,0) + \hat{f}(0,1) - \hat{f}(0,0) - \hat{f}(1,1) \geq 0$$

and proves the claim. $\square$

Example A.4 We are going to demostrate the proof given above for a function $f \in V_2^3$. Let f be a two-regular function and pick a $(0;2,1)$ partition $\{\varnothing;\{1,2\},\{3\}\}$ of I_3. Following the notation described in the proof, $\bar{f} \in V_2^2$ and $\tilde{f} \in V_2^2$, defined as

$$\bar{f}(x_1,x_2) = f(1,x_1,x_2)$$
$$\tilde{f}(x_1,x_2) = f(x_1,0,x_2)$$

are regular, hence

$$\bar{f}(0,1) + \bar{f}(1,0) - \bar{f}(0,0) - \bar{f}(1,1) \geq 0$$

implies

$$f(1,0,1) + f(1,1,0) - f(1,0,0) - f(1,1,1) \geq 0 \tag{A.3}$$

and

$$\tilde{f}(0,1) + \tilde{f}(1,0) - \tilde{f}(0,0) - \tilde{f}(1,1) \geq 0$$

implies

$$f(0,0,1) + f(1,0,0) - f(0,0,0) - f(1,0,1) \geq 0 \tag{A.4}$$

addition of Inequalities (A.3) and (A.3) gives

$$f(1,1,0) + f(0,0,1) - f(1,1,1) - f(0,0,0) \geq 0$$

which means

$$\hat{f}(0,1) + \hat{f}(1,0) - \hat{f}(0,0) - \hat{f}(1,1) \geq 0$$

The same approach holds for any partition of I_3. $\square$

APPENDIX B

The proof of the Proposition 3.19

Let us restate the Proposition 3.19.

Proposition B.1 *The family V_2^n with the relation $\geq$ given by Definition 3.17 is a partially ordered set [23], i.e.*

$$\begin{aligned} i. &\quad f \geq f \\ ii. &\quad g \geq f \textit{ and } f \geq g \implies g = f \\ iii. &\quad g \geq f \textit{ and } f \geq h \implies g \geq h \end{aligned}$$

for any $f, g, h \in V_2^n$.

Proof The conditions i and iii are immediate. For ii, let $b = (b_1, .., b_n) \in \Lambda_2^n$ and define the partition $\{\alpha^0; \alpha^1\}$ of I_n as follows

$$\alpha^0 = \{\alpha_1^0, .., \alpha_p^0\} = \{i \in I_n : b_i = 0\}$$
$$\alpha^1 = I_n \setminus \alpha^0$$

where of course p is the number of zeros in b. Define $a = (a_1, .., a_p) \in \Lambda_2^p$ such that $a_i = 0$ for $i = 1, .., p$. Then obviously we have

$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(f)(1) = f(b_1, .., b_n)$$
$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(g)(1) = g(b_1, .., b_n)$$

and

$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(f)(0) = \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_p)}(g)(0) = f(0, .., 0) = g(0, .., 0) = 0$$

for any $f, g \in V_2^n$. Therefore $g \geq f$ implies

$$\mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(g)(1) \geq \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(0) - \mathcal{P}^{(a_1..a_p)}_{\alpha^0;\alpha^1}(f)(1)$$

and as a result

$$-g(b_1, .., b_n) \geq -f(b_1, .., b_n)$$

On the other hand $f \geq g$ similarly implies

$$-f(b_1, .., b_n) \geq -g(b_1, .., b_n)$$

Thus $g \geq f$ and $f \geq g$ implies $f(b_1, .., b_n) = g(b_1, .., b_n)$ and proves the claim. $\square$

APPENDIX C

The proof of the Proposition 3.20

Let us first restate the Proposition 3.20.

Proposition C.1 *Let $f, g \in V_2^n$. If for any $(n-1;1)$ partition $\{\alpha^0;\alpha^1\}$ of I_n and for any $a = (a_1, .., a_{n-1}) \in \Lambda_2^{n-1}$ we have*

$$\mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(g)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(g)(1) \geq \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(f)(0) - \mathcal{P}_{\alpha^0;\alpha^1}^{(a_1..a_{n-1})}(f)(1)$$

then $g \geq f$.

Proof Let $\{\beta^0;\beta^1\}$ be a $(p;q)$ partition of I_n and $b = (b_1, .., b_p) \in \Lambda_2^p$. For $q = 0, 1$ there is nothing to prove so let $q \geq 2$. For $j = 1, .., q$, define

$$\begin{aligned} \gamma^{0j} &= \{\beta_1^1, .., \beta_{j-1}^1, \beta_1^0, .., \beta_p^0, \beta_{j+1}^1, .., \beta_q^1\} \\ \gamma^{1j} &= \{\beta_j^1\} \end{aligned}$$

and

$$\begin{aligned} c^j &= (c_1^j, .., c_{n-1}^j) \\ &= (\underbrace{1, .., 1}_{j-1 \text{ times}}, b_1, .., b_p, \underbrace{0, .., 0}_{q-j \text{ times}}) \end{aligned}$$

Define

$$\begin{aligned} \hat{f}_j &= \mathcal{P}_{\gamma^{0j};\gamma^{1j}}^{(c_1^j..c_{n-1}^j)}(f) \\ \hat{g}_j &= \mathcal{P}_{\gamma^{0j};\gamma^{1j}}^{(c_1^j..c_{n-1}^j)}(g) \end{aligned}$$

Note that

$$\hat{f}_1(0) = \mathcal{P}_{\beta^0;\beta^1}^{(b_1,..,b_p)}(f)(0) \qquad \hat{f}_q(1) = \mathcal{P}_{\beta^0;\beta^1}^{(b_1,..,b_p)}(f)(1)$$

$$\hat{g}_1(0) = \mathcal{P}_{\beta^0;\beta^1}^{(b_1,..,b_p)}(g)(0) \qquad \hat{g}_q(1) = \mathcal{P}_{\beta^0;\beta^1}^{(b_1,..,b_p)}(g)(1)$$

Notice that for $j = 1, .., q-1$, $\hat{f}_j(1) = \hat{f}_{j+1}(0)$ which implies

$$\sum_{j=1}^{q} (\hat{f}_j(0) - \hat{f}_j(1)) = \hat{f}_1(0) - \hat{f}_q(1)$$
$$= \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(f)(0) - \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(f)(1)$$

Similarly since $\hat{g}_j(1) = \hat{g}_{j+1}(0)$ for $j = 1, .., q-1$, we have

$$\sum_{j=1}^{q} (\hat{g}_j(0) - \hat{g}_j(1)) = \hat{g}_1(0) - \hat{g}_q(1)$$
$$= \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(g)(0) - \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(g)(1)$$

Therefore since the hypothesis of the proposition implies that

$$\hat{g}_j(0) - \hat{g}_j(1) \geq \hat{f}_j(0) - \hat{f}_j(1)$$

for any $j = 1, .., q$, we have

$$\mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(g)(0) - \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(g)(1) \geq \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(f)(0) - \mathcal{P}^{(b_1..b_p)}_{\beta^0;\beta^1}(f)(1)$$

which finishes the proof. □

Example C.2 This example demonstrates the proof given above for a simple case. Let $f, g \in V_2^5$, $p = 3$, $q = 2$ and $b = (b_1, b_2, b_3)$. Let $\{\beta^0; \beta^1\}$ be a $(p; q)$ partition of I_5, where $\beta^0 = \{1, 3, 4\}$ and $\beta^1 = \{2, 5\}$. Then

$$\gamma^{01} = \{1, 3, 4, 5\} \qquad \gamma^{02} = \{2, 1, 3, 4\}$$
$$\gamma^{11} = \{2\} \qquad \gamma^{12} = \{5\}$$

and $c^1 = (b_1, b_2, b_3, 0)$, $c^1 = (1, b_1, b_2, b_3)$. Note that

$$\mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(f)(0) = f(b_1, 0, b_2, b_3, 0) = \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(f)(0)$$
$$\mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(f)(1) = f(b_1, 1, b_2, b_3, 1) = \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(f)(1)$$

and similarly

$$\mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(g)(0) = g(b_1, 0, b_2, b_3, 0) = \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(g)(0)$$
$$\mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(g)(1) = g(b_1, 1, b_2, b_3, 1) = \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(g)(1)$$

Note moreover that

$$\mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(f)(1) = f(b_1,1,b_2,b_3,0) = \mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(f)(0)$$
$$\mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(g)(1) = g(b_1,1,b_2,b_3,0) = \mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(g)(0)$$

The hypothesis of the proposition gives

$$\mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(g)(0) - \mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(g)(1) \geq \mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(f)(0) - \mathcal{P}^{(b_1b_2b_30)}_{\gamma^{01};\gamma^{11}}(f)(1)$$
$$\mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(g)(0) - \mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(g)(1) \geq \mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(f)(0) - \mathcal{P}^{(1b_1b_2b_3)}_{\gamma^{02};\gamma^{12}}(f)(1)$$

thus

$$g(b_1,0,b_2,b_3,0) - g(b_1,1,b_2,b_3,0) \geq f(b_1,0,b_2,b_3,0) - f(b_1,1,b_2,b_3,0)$$
$$g(b_1,1,b_2,b_3,0) - g(b_1,1,b_2,b_3,1) \geq f(b_1,1,b_2,b_3,0) - f(b_1,1,b_2,b_3,1)$$

and adding them up gives

$$\mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(g)(0) - \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(g)(1) \geq \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(f)(0) - \mathcal{P}^{(b_1b_2b_3)}_{\beta^0;\beta^1}(f)(1)$$

The same procedure applies for any partition of I_n and any sequence b of binary numbers. $\square$

REFERENCES

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

[2] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, 2006.

[3] D. P. Bertsekas. *Network Optimization, Continuous and Discrete Models*. Athena Scientific, 1998.

[4] J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society, B*, 48(3):259–302, 1986.

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–37, 2004.

[6] Y. Boykov, O. Veksler and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11):1222–1239, 2001.

[7] P. Brémaud. *Markov Chains - Gibbs Fields, Monte Carlo Simulation and Queues*. Springer, 1999.

[8] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

[9] T. F. Chan and J. Shen. *Image Processing and Analysis, Variational, PDE, Wavelet and Stochastic Methods*. SIAM, 2005.

[10] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *J. Math. Imaging Vis.*, 26(3):261–276, 2006.

[11] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part ii: Levelable functions, convex priors and non-convex cases. *J. Math. Imaging Vis.*, 26(3):277–291, 2006.

[12] A. Foi, V. Katkovnik, K. Egiazarian and J. Astola. A novel anisotropic local polynomial estimator based on directional multiscale optimizations. In *Proceedings of the 6th IMA International Conference on Mathematics in*

Signal Processing, volume 3, pages 79–82, Cirencester, UK, 2004. Available at http://www.cs.tut.fi/lasip/results2D.

[13] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 939–946, 2005.

[14] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6):721–741, 1984.

[15] D. Gleich. Matlab bgl v2.0. Technical report, Stanford University, Institute for Computational and Mathematical Engineering, 2006.

[16] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. Technical report, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA, 2007.

[17] D. M. Greig, B. T. Porteous and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Soc., Series B*, 51(2):271–279, 1989.

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

[19] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(10):1333–1336, 2003.

[20] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2005.

[21] P. Kohli, M.P. Kumar and P.H.S Torr. P3 and beyond: Solving energies with higher order cliques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[22] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, 2004.

[23] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, 1989.

[24] S. Li. *Markov Random Field Modelling in Computer Vision*. Springer, 1995.

[25] Stan Z. Li. Markov random field models in computer vision. In *ECCV (2)*, pages 361–370, 1994.

[26] Z. Liao and J. Zhao. Exact optimization for a class of second order markov random field via graph cuts. In *Proceedings of the Fourth IIIE International Conference on Machine Learning and Cybernetics*, pages 5512–5516, Guangzhou, China, 2005.

[27] P. Pérez. Markov random fields and images. *CWI Quarterly*, 11(4):413–437, 1998.

[28] J. C. Picard and H. D. Ratliff. Minimum cuts and related problems. *Networks*, 5:357–370, 1975.

[29] S. Roy. Stereo without epipolar lines: A maximum-flow formulation. *Int. J. Comput. Vision*, 34(2-3):147–161, 1999.

[30] Jeremy G. Siek, Lie-Quan Lee and Andrew Lumsdaine. *The Boost Graph Library User Guide and Reference Manual (With CD-ROM)*. Addison-Wesley Professional, 2001.

[31] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. F. Tappen and C. Rother. A comparative study of energy minimization methods for markov random fields. In *Proceedings of the 9th European Conference on Computer Vision*, volume 2, pages 16–29, Graz, Austria, 2006.

[32] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Springer, 2000.

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

[34] B. Zalesky. Efficient determination of gibbs estimators with submodular energy functions, 2003. Available at http://arxiv.org/abs/math/0304041v1.

VITA

PERSONAL INFORMATION

Surname, Name: Karcı, Mehmet Haydar

Nationality: Turkish (TC)

Date and Place of Birth: 2 February 1975 , Ankara

Marital Status: Single

Phone: +90 542 477 46 05

email: thaydart@yahoo.com

EDUCATION

Degree	Institution; Year of Graduation
MS	METU, Electrical and Electronics Engineering; 2000
BS	METU, Electrical and Electronics Engineering; 1997
High School	Ankara Science High School; 1993

PROFFESSIONAL EXPERIENCE

Year	Institution; Status
1997–2004	Tübitak, Bilten; Researcher
2004–2005	Visioprime; Engineer