M.S. Thesis

Stochastic Graduated Graph Approximation Algorithm for MRF optimization

확률론적 순차적 그래프 근사를 이용한
MRF 최적화

BY

Sergei Liubich

February 2014

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

Markov random fields have been powerful models in computer vision but tractable algorithms to obtain exact solution for the corresponding energy functions are limited; approximate solutions, in most cases are provided for efficiency. In this work graduated optimization technique is applied in a novel way to develop an efficient algorithm for solving general multi-label MRF optimization problem called Stochastic Graduated graph approximation (SGGA) algorithm. The algorithm initially minimizes a simplified function and progressively transforms that function until it is equivalent to the original function. However, it is hard to find how to generate the sequence of intermediate functions and what parameter to use for making transition from one problem to another. For this we propose a new iterative method of building the sequence of approximations for the original energy function. We exploit a stochastic method to generate intermediate functions, which guides the intermediate solutions to the near-optimal solution for the original problem. The transition from one intermediate problem to another is controlled by the schedule of gradual addition of edges. In each iteration, a deterministic algorithm such as block ICM is applied to minimize intermediate functions and to generate initial solution for the next problem. The proposed algorithm guarantees the convergence of local minimum. We test on a synthetic image deconvolution problem and also on the set of experiments with the OpenGM2 benchmark.

Key words: MRF, discrete optimization, graph approximation.

Student number: 2013-22506

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of research

Markov random fields (MRFs) have attracted much attention in computer vision for recent decades and have been successfully applied to many vision applications [2,3]: segmentation, stereo matching, denoising, inpainting and more. Generally, MRF models are formulated as follows. Given a graph $G = \{\mathcal{V}, \mathcal{E}\}$ consisting of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$, a random variable x_i associated with each node $i \in \mathcal{V}$ takes a value from a set of labels $\Lambda = \{1, 2, \cdots, L\}$. Let ψ_i be unary potential functions and ψ_{ij} be pairwise potential functions defined on nodes and edges. The energy function of the pairwise MRF is given by

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{1.1}$$

for the set of random variables $\mathbf{x}$. The Maximum a Posteriori (MAP) problem of the joint probability distribution can be transformed to a minimization problem of the function (4.1).

The Hammersley – Clifford theorem establishes that the joint probability of any MRF can be represented by Gibbs distribution. It is so called Markov – Gibbs equivalence. The Gibbs distribution of MRF $\mathbf{x}$, defined on the graph G, with the neighboring system $\mathcal{E}$, is given by

$$P(\mathbf{x}) = \frac{1}{Z} \prod_j \psi_j(x_j), \tag{1.2}$$

where Z is a normalizing constant and $\psi_j(x_j)$ is a clique potential function defined on the set of random variables $\mathbf{x}$ for clique j .

Computer vision problems have used MRF to formulate the probability function for possible solutions and achieve the most probable solution. That is to find the MAP solution from the given probability function. The MAP solution is defined by

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmax}} P(\mathbf{x}) \tag{1.3}$$

The energy function (4.1) is connected with the joint probability function by the following relation

$$E(\mathbf{x}) = -lnP(\mathbf{x}) + Z, \tag{1.4}$$

$$E(\mathbf{x}) = \sum_i \psi_i(x_i), \tag{1.5}$$

where

$$\psi_i(\cdot) = -ln\psi_i(\cdot) \tag{1.6}$$

Now the MAP solution $\mathbf{x}^\star$ can be represented by the energy function as follows

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmin}} E(\mathbf{x}) \tag{1.7}$$

Therefore, estimating the MAP solution for the given MRF is equivalent to finding the solution $\mathbf{x}^\star$ which minimizes the energy function (4.1).

Obtaining the global minimum of the energy function (1) is a NP – hard problem; only few cases can be solved exactly in polynomial time. For example, message passing algorithms, such as belief propagation (BP) [4] and tree-reweighted message passing (TRW) [5] provide exact solution for trees and chains. Graph cuts [6, 7] can exactly minimize binary submodular energy functions even in a loopy structure. But most cases resort to approximate solutions.

For general MRF optimization problem, some algorithms have obtained only approximate solutions in deterministic search approaches. Move – making algorithms – iterated conditional modes [8], α – expansion and $\alpha\beta$ – swap [6], iteratively make local moves towards lower energy state. That is, the algorithms simplify an inference problem to a series of reduced problems by restricting a label space in which they maintain the current best solution until they cannot find further minimizing proposals. Thus the deterministic methods are efficient for searching solutions but get stuck to local optimum.

Stochastic approaches have also been applied to minimize energy functions of MRF problems. The Markov chain Monte Carlo (MCMC) is an important class of stochastic methods for the aforementioned problem. It takes samples from a probability distribution based on Markov chain and approximates the target distribution. Since the advent of Metropolis – Hastings method [9], the MCMC became a convenient tool that provides simple procedures to design the kernel that converges to the desired stationary distribution. Gibbsian simulated annealing (Gibbsian SA) [10] is a general optimization method using the Gibbs sampling, which is simplified MCMC approach. These sampling based algorithms can reach the global optimal solution in

theory because stochastic methods are able to avoid getting stuck in local minimum. However, despite its simplicity and theoretical elegance, they are not widely used in the MRF optimization because of its slow convergence.

Therefore, recent works [11, 12] have focused on methods of improving MCMC's finite – time behavior. In [11] they investigated two ideas, breaking detailed balance and updating multiple nodes at a time to overcome the main obstacle of slow convergence of MCMC – based algorithms, which significantly improved the convergence speed of Markov chain. Kim and Lee [12] also have combined stochastic and deterministic algorithms and shown the effectiveness of this approach in solving challenging MRF problems. This approach achieves lower energy solution than other sampling – based and deterministic methods. However, the proposed methods are still slower than deterministic algorithms to get lower energies.

1.2 Objective

In this work, we propose a new efficient algorithm for solving general multi-label MRF optimization problem. We combine advantages of stochastic and deterministic methods. Instead of combining MCMC with deterministic methods, we apply another powerful optimization technique, Graduated non-convexity (GNC) [13]. GNC guides serial local solutions to the global optimum by approximating a complex function. However, the approximation method has not been used widely due to the difficulty of application to specific problems. We propose stochastic graduated graph approximation algorithm (SGGA) for solving general discrete MRF problems. We generate a series of approximate energy functions by stochastically selecting edges from an original graph and guide to lower energy state, which is usually close to the

global optimum. Here we show that our algorithm:

- *Achieves lower energy than conventional deterministic optimization algorithms and efficient hybrid MCMC-based algorithm from [12], MCMC-GD, in solving image deconvolution problem and*
- *faster convergence than MCMC-GD and similar to deterministic algorithms. As far as we know, this is the first work based on the concept of graduated optimization for solving discrete MRF optimization problems. We compare our competitive results for real applications from the OpenGM2 benchmark with various deterministic methods and with recently published fast MCMC-based algorithm [11].*

1.3 Outline of thesis

In this work graduated optimization technique is applied in a novel way to develop an efficient algorithm for solving general multi-label MRF optimization problem called Stochastic Graduated graph approximation (SGGA) algorithm.

In Chapter 2, related works are discussed. SGGA algorithm is based on the concept of graduated optimization. According to our knowledge there are not many papers devoted to discrete MRF optimization which apply this concept. Graduated non – convexity (GNC) and sequential Monte Carlo (SMC) are similar to our algorithm. Basic ideas of these methods are described.

In Chapter 3, we introduce a new method for guiding a local search. Our method is based on the graduated graph approximation. The overall algorithm (SGGA) and it's modification are introduced at the end of this chapter.

In Chapter 4, we describe the block ICM method which is applied for minimizing intermediate energy functions in our iterative algorithm. Also we describe the dynamic programming (DP) algorithm which is applied within block ICM in order to efficiently update a labeling assignment on a set of nodes in approximated graphs.

In Chapter 5, we evaluate efficiency of SGGA algorithm on a synthetic experiment and OpenGM2 benchmark.

In Chapter 6, we make conclusions about our research work.

Chapter 2

Related works

2.1 Graduated optimization

Graduated optimization is a global optimization technique that tries to solve a difficult optimization problem by initially solving a simplified problem, and gradually transforming that problem (while optimizing intermediate problems) until it is equivalent to the difficult optimization problem.

Graduated optimization is an improvement to hill climbing that enables a hill climber to avoid settling into local optima. It decompose a difficult optimization problem into a sequence of optimization problems, such that the first problem in the sequence is convex (or nearly convex), the solution to each problem gives a good initial point to the next problem in the sequence, and the last problem in the sequence is the difficult optimization problem that it ultimately seeks to solve. In practice, graduated optimization gives better results than simple hill climbing. Further, when certain conditions exist, it can be shown to find an optimal solution to the final problem in the sequence [13]. These conditions are:

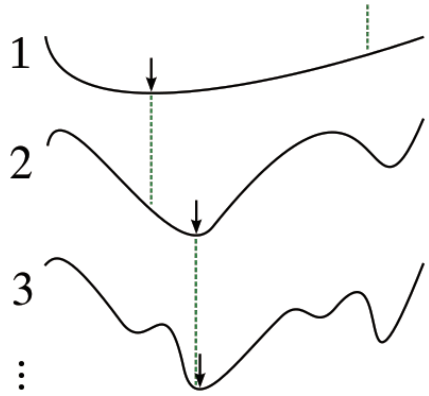


Figure 2.1: Graduated non-convexity method: it breaks a difficult optimization problem into a sequence of optimization problems, such that the first problem in the sequence is convex (or nearly convex), the solution to each problem gives a good starting point to the next problem in the sequence, and the last problem in the sequence is the difficult optimization problem that it ultimately seeks to solve.

- *the first optimization problem in the sequence can be solved exactly given the initial point;*
- *the locally convex region around the global optimum of each problem in the sequence includes the point that corresponds to the global optimum of the previous problem in the sequence.*

It can be shown inductively that if these conditions are met, then a hill climber will arrive at the global optimum for the difficult problem. Unfortunately, it can be difficult to find a sequence of optimization problems that meet these conditions. Often, graduated optimization yields good results even when the sequence of problems cannot be proven to strictly meet all of these conditions.

Blake and Zisserman [13] have developed the deterministic and approximate ap-

proach based on the concept of graduated optimization, Graduated non – convexity (GNC). This method implies approximation of a general non-convex function by simplified functions, e.g. a convex function. This approximation slowly varies towards the original function, in the hope that the guided local minimum will converge to the global minimum in the original solution space (see Figure 2.1). However, they found the approximation only in special cases, such as a weak string model.

In [14], they extended to a general case and showed that GNC was faster than Simulated annealing. Terzopoulos and Witkin [15, 16] also proposed continuation optimization methods similar to GNC.

The issue is that only a general scheme for creating GNC algorithms has been discussed. For example, any stabilizing term, which is determined as the sum of functions of the local derivatives of the solution, can be approximated by Gaussian scale space extension, yielding a convex solution space. By slowly varying the standard deviation of the Gaussian towards zero, the solution space of the approximation will slowly vary towards the non-convex solution space. In computer vision, some applications have been exploited in image deblurring [17], shape matching [18], image segmentation [19], image denoising [20], template matching [21], Hough transform [22], edge detection [23], early vision [24] and image matching [25]. However, application of GNC method is very limited as no known principle for generating between approximated ones exist.

Besides, as far as we know, graduated optimization has not been applied for solving discrete MRF optimization problems.

2.2 Sequential Monte Carlo

There is another optimization method related to our algorithm, i.e. Sequential Monte Carlo (SMC) [21].

The SMC methods (e.g. particle filter) [26] are widely used for dynamic problems in computer vision, such as object tracking [27]. The main idea of the SMC is to sequentially sample the target probability distribution π_t at time t with a set of particles, that is a group of weighted random samples. A weight $w_t^{(i)}$ represents the importance of the associated sample $x_t^{(i)}$.

The samples $x_t^{(i)}$ at a time t are generated from the importance distribution ν_t, which is constructed by a Markov transition kernel K and the previous distribution π_{t-1}; each sample $x_t^{(i)}$ evolves from $x_{t-1}^{(i)}$ according to the kernel K , and the importance distribution ν_t is represented by

$$\nu_t(x_t) = \sum_{x_{t-1}} \pi_{t-1}(x_{t-1})K(x_{t-1}, x_t). \tag{2.1}$$

The weight $w_t^{(i)}$ of each sample $x_t^{(i)}$ is proportional to the ratio of the target distribution π_t to the importance distribution ν_t, also is defined as

$$W_t^{(i)} = \frac{\pi_t(x_t^{(i)})}{\nu_t(x_t^{(i)})}, \tag{2.2}$$

$$w_t^{(i)} = \frac{W_t^{(i)}}{\sum_{i=1}^{N} W_t^{(i)}}. \tag{2.3}$$

Then, the particles as the weighted samples asymptotically converge to the target distribution:

$$\pi'_t(x_t) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(x_t) \underset{a.s.}{\rightarrow} \pi_t(x_t), N \rightarrow \infty, \tag{2.4}$$

where δ denotes the Dirac delta function.

SMC has been applied to graph matching problem [28] in which a sequence of intermediate distributions are introduced from a simple initial distribution and gradually moving towards the final target distribution equivalent to the original objective function of the problem.

Our algorithm initially minimizes a simplified energy function and gradually transforms that function until it is equivalent to the original energy function. In our case this transformation is performed via graduated graph approximation. Transition from one function to another is performed by random addition of edges. We get a new solution by randomly generating a new intermediate graph and optimizing corresponding energy function by block ICM. It can be interpreted as sampling from intermediate distribution. Our approach differs from SMC in that we do not increase dimension of the problem, instead, we increase complexity of the problem. Also we cannot generate a set of solutions at a time and choose the best one as SMC, because in that case we need to generate multiple structures in each iteration and optimize each of them, that is computationally very expensive.

Like Simulated annealing [10], our random selection of edges produces solutions that allow some uphill moves in energy state, in which the sequence of solutions ranges in long and gradually in short distance. Instead, our algorithm accepts a new solution with probability 1 and does change the structure of the function.

Chapter 3

Stochastic graduated graph approximation

3.1 Graph approximation by scanlines

The main idea of the graduated optimization is to obtain an optimal solution, which is aimed to be the global optimum, through intermediate functions simplified from an original function. For example, the outer bound of a non-convex function is approximated as a convex function and considered to get a minimum value near the global optimum. We initially minimize the simplest function of the original one and progressively generate complex functions gradually until it is equivalent to the original function. Meanwhile, minimizing the intermediate ones makes the solutions move towards the global optimum. Thus, the graduated optimization technique aims to guide the solutions through the sequence of intermediate optimization problems to obtain a good solution for the original problem. The issue here is how to generate the sequence of intermediate functions to guide the solutions. For this, we propose

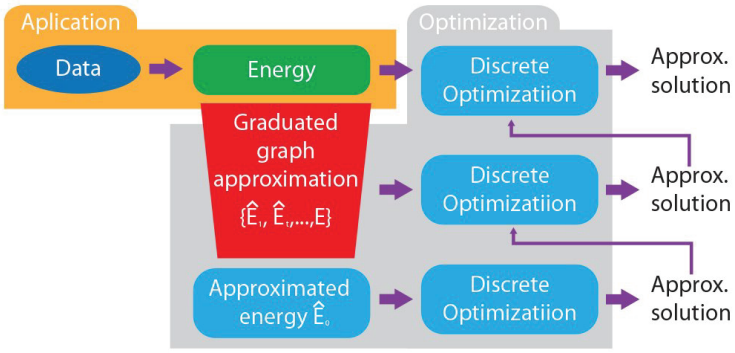


Figure 3.1: Unified framework: we propose a new iterative method of building the sequence of approximations for the original energy function. We exploit a stochastic method to generate intermediate functions, which guides the intermediate solutions to the near-optimal solution for the original problem. Our framework is unified in the sense that different problems with different energies share the same approximation scheme, making our framework widely applicable and general. (The scheme has been borrowed from [1].)

a new iterative method of building the sequence of approximations for the original energy function (see Figure 3.1).

For simplicity, we consider a grid graph G with horizontal and vertical edges between nodes. We exploit edge addition for obtaining approximated functions. This simple approximation is applicable to any class of energy functions. Let an approximated graph denoted by $\widetilde{G}$, where $\widetilde{\mathcal{E}}$ is a subset of $\mathcal{E}$. All the nodes $\mathcal{V}$ are maintained for the same structure and the corresponding potentials, i.e. ψ_i and ψ_{ij} are inherited from the energy function (4.1). Figure 3.2 shows a sequence of graphs are generated by gradually adding edges to a set of chains (or scanlines) in row and column ways.

By increasing the size of $\widetilde{\mathcal{E}}$, the intermediate graphs $\widetilde{G}$ become complex and

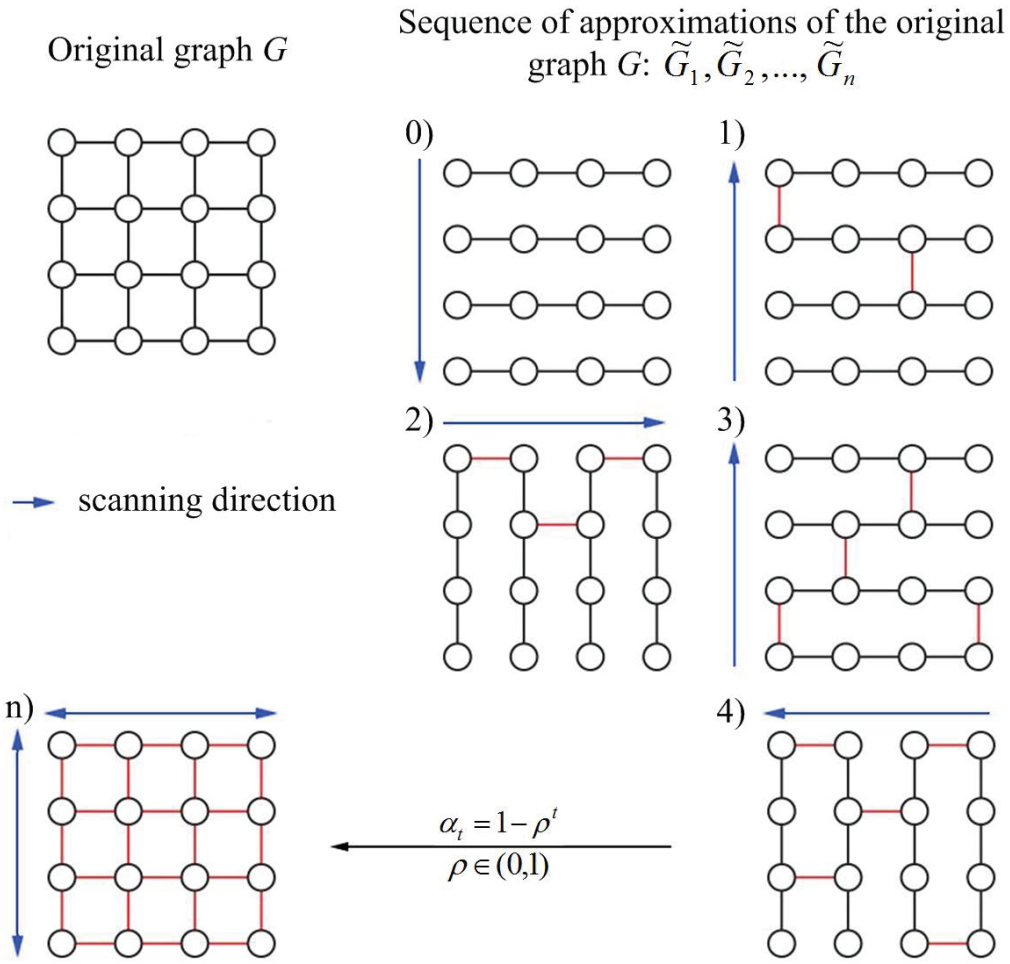


Figure 3.2: Graduated graph approximation: the basic idea of the overall algorithm. SGGA builds the sequence of approximated graphs $\Gamma = \{\widetilde{G_0}, \widetilde{G_1}, ..., \widetilde{G_t}, ..., G\}$ and optimizes corresponding energy functions. Optimization of each approximated energy function is performed by block ICM-based local search method. Scanning generates initial solution for each subsequent problem. Diversity of proposed initial solutions is achieved by random addition of edges to approximated graphs. Number of added edges in each iteration is controlled by user-specified parameter α.

equivalent to the graph G . Thus, we generate a sequence of intermediate graphs $\Gamma = \{\widetilde{G_0}, \widetilde{G_1}, ..., \widetilde{G_t}, ..., G\}$ with the corresponding energy functions $\Phi = \{\widetilde{\mathcal{E}_0}, \widetilde{\mathcal{E}_1}, ..., \widetilde{\mathcal{E}_t}, \mathcal{E}\}$, where an approximated function $\widetilde{\mathcal{E}_t}$ is formulated as follows:

$$\widetilde{\mathcal{E}_t}(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \widetilde{\mathcal{E}}} \psi_{ij}(x_i, x_j) \tag{3.1}$$

Note that we assume the initial graph $\widetilde{G_0}$ consists of a set of disjoint chains and increase the subset of edge $\widetilde{\mathcal{E}}$ by adding only edges between the initial chains.

Transition between a series of intermediate graphs is performed by gradual addition of edges between rows and columns. The schedule of adding edges at each iteration is an important control parameter in our framework. We increase the number of active edges by the proportion at iteration t

$$\alpha_t = 1 - \rho^t, \tag{3.2}$$

where $\rho \in (0, 1)$ is a step size.As extreme cases, when $\rho = 1$, the problem becomes a scanline approximation, which is performed by row-based and column-based alternation. This type of approximation has been proposed in early stereo matching, denoising problems and more [11, 29–33]. When $\rho = 0$, the graph is equivalent to the original target graph for which we can exploit any optimization techniques like move-making algorithms. Therefore, this scheduling recovers the original graph gradually.

For optimizing each intermediate energy function $\widetilde{\mathcal{E}_t}$, associated with graph $\widetilde{G_t}$ depicted in Figure 3.2, dynamic programming algorithm is subsequently applied to all scanlines in graph $\widetilde{G_t}$. We summarize all steps described so far in the following pseudo code of the overall algorithm (3.1).

Algorithm 1 Stochastic Graduated Graph Approximation Algorithm

Set iteration count $t = 0$, $\alpha_t = 1$ and $\rho \in (0, 1)$

Initialize the initial assignment $\mathbf{x_0}$

for Each scanline direction **do**

Generate a graph $\widetilde{G}_t$ according to α_t

$\mathbf{x}_{t+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \widetilde{E_t}(\mathbf{x})$ by block ICM and DP

Update $\alpha_t = \rho\alpha_{t-1}$

if the maximum iteration t or $\alpha_t \approx 0$ **then**

return the current $\mathbf{x}$

end if

end for

Our algorithm scans rows from top to bottom, then columns from left to right, then rows from bottom to top and then columns from right to left. Thus, influence of each node on others is propagated along scanlines in possible directions. This process has been called scanning. The order of scanning directions can be changed without degradation of performance of the overall algorithm. For instance, this method preserves vertical and horizontal consistency of an image. In the last iteration, SGGA solves the original problem, i.e. minimizes the energy function which corresponds to the graph G but it uses a preoptimized configuration obtained by the sequence of previous iterations. In the last iteration the reconstructed graph is the original graph. Using the block ICM algorithm, the convergence of SGGA is guaranteed.

However, given the number of active edges for a graph $\widetilde{\mathcal{E}}_t(\mathbf{x})$, we select the active edges randomly at each iteration. This random selection helps guided solutions pervaded in a wide range between iterations and makes long – range movement in

solution space. In addition, the varying number of active edges controls complexity of the optimization problem and also range of move – making proposals. This is, as more edges are added, the connected nodes are strongly conditioned on each other. This makes move – making small. On the contrary, the movement becomes large in solution space.

3.2 Graph approximation by trees

There have been other approaches to approximate the original function in restricted structures. Some structures are known to be tractable, such as bounded treewidth subgraphs (e.g. tree and outer-planar graph) [32, 34–36].

Instead of updating scanline at a time, we may choose a tree structure. However, we experimentally have found out it is not good idea. Another version of SGGA has been developed. It is based on iterative random generating of tree structures. It has been called Stochastic Tree-based Graph Approximation algorithm (STGA)(2).

A single row (or a single column) in SGGA is substituted by a tree. For each iteration we randomly generate trees. At each iteration we randomly choose one node to be a root. Then we examine all the neighboring nodes in random order until all nodes are examined. If adding a node make a cycle, we throw it out. If adding a node is ok, we add that node to the current tree. Thus, a random tree is built. After generating a tree we randomly delete edges between two nodes: one node is inside the tree, another one is outside. At first iteration we delete all edges, at the last - none. In order to optimize energy function associated with a tree we apply general DP algorithm. Experimentally we found out that a tree cannot cover more than 50% of nodes and therefore information about configuration of nodes might be

Algorithm 2 Stochastic Tree-based Graph Approximation algorithm

Set iteration count $t = 0$, $\alpha_t = 1$ and $\rho \in (0, 1)$

Initialize the initial assignment $\mathbf{x_0}$

FOR $t = 0$ to N

Generate a tree $\widetilde{T}_t$ according to α_t

$\mathbf{x}_{t+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \widetilde{E_t}(\mathbf{x})$ by general DP

Update $\alpha_t = \rho\alpha_{t-1}$

if the maximum iteration t or $\alpha_t \approx 0$ **then**

return the current $\mathbf{x}$

end if

ENDFOR

lost.

Despite the fact that using tree structure we can update more nodes at a time, it turns out to be less efficient than scanline approach. The inappropriateness of these structured approximations to the proposed algorithm can be attributed to the following reason. By constructing a tree, the distance between two nodes is distorted. Although two nodes could be close to each other in the original graph, they can be apart through a detour generated by tree structure. And although close nodes are more correlated, they influence on each other less in a tree structure than in a scanline. Therefore, embedding restricted structures to our framework is not appropriate.

Chapter 4

Minimization of intermediate energy functions

4.1 Block Iterated conditional modes

4.1.1 Block Iterated conditional modes: general idea

Block Iterated conditional modes (block ICM) is a generalization of ICM. ICM algorithm can be applied to any given discrete factor graph but has been originally proposed for images by Besag [8].

ICM uses the following property: if all but one variable were observed, then solving for the MAP state of the single dependent variable would be easy. The algorithm iteratively updates one variable at a time, keeping all other variables fixed.

This is a local search method with neighborhood relation:

$$N_p(\mathbf{x}) = \{x_1, x_2, ..., x_{p-1}, y_p, x_{p+1}, ..., x_p | y_p \in \mathbf{x}_p\}, \tag{4.1}$$

where $p = 1, .., P$ and $P = |\mathcal{V}|$ indexes the dependent variables in the model. Iterating over all the neighborhoods, the values of variables y_p are effectively optimized one – by – one, improving the overall objective function. The solution returned by ICM guarantees local optimality with respect to the neighborhoods. The larger the neighborhood, the stronger this local optimality guarantee. The largest neighborhood – the original set of nodes itself, recovers the original problem. In general, it is desired to select the largest neighborhood relation that still allows for efficient optimization.

In block ICM algorithm the neighborhood is enlarged by a larger subset of variables for updating. Fixing a subset of variables to their current values corresponds to conditioning the probability distribution. In other words, the neighborhood is defined by the resulting conditioned distribution. Optimization inside the neighborhood is efficiently solvable as long as the subset of variables forms a tree – structured subgraph in the original graph. Whereas the original ICM neighborhood is as large as the number of labels the variable can take, the search space explored by the block ICM neighborhoods is exponential in the number of variables optimized over. For grid – structured graphs a typical subset of variables induced by chains, i.e. scanlines.

4.1.2 Block ICM for graduated graph approximation

In Chapter 3, we have generated a set of intermediate functions Φ. Now we can perform minimization of the approximated functions. Many existing algorithms are applicable for this purpose but the performance depends on local optimization method that is applied in each iteration. We exploit efficient move-making algorithms, i.e. block ICM –based method to optimize each energy function $\widetilde{\mathcal{E}}_t(\mathbf{x})$. Especially, the

performance of local search algorithms such as move – making algorithms depends on the initial values, which enables our algorithm to guide intermediate solutions to the target solution naturally. For example, the current solution $\mathbf{x}_t$ from $\widetilde{\mathcal{E}}_t(\mathbf{x})$ is used for the initial state of the next energy function $\widetilde{\mathcal{E}}_{t+1}(\mathbf{x})$ for initialization.

In our formulation, the graph $\widetilde{G_t}$ corresponding to $\widetilde{\mathcal{E}}_t(\mathbf{x})$ is represented by a disjoint set of scanlines $S_s = \{V_s, \mathcal{E}_s\}$, where $\mathcal{V}_s$ is the subset of nodes and $\mathcal{E}_s$ is the associated edges respectively (see iteration 0 in Figure 3.2). The union of scanlines should include all the nodes $\mathcal{V}$, i.e. $\cup_s \mathcal{V}_s = \mathcal{V}$. One scanline S_s is selected from the graph $\widetilde{G_t}$ at a time conditioned on the remaining variables and edges in the graph. Figure 3.2 describes the procedures of scanning order. The scanning order of block ICM is 4 directional and the basic blocks of scanlines are considered in row or column way depending on the direction. Optimization on a block of nodes can be exactly performed by DP in polynomial time. Other structures on subsets of nodes, e.g. tree – structures, can be considered but the performance – computation time, exact solution and move-making range - varies. We choose a chain structure for efficiency and tractability of exact solution by DP.

Although, block ICM does not guarantee convergence to the MAP but, providing that the algorithm starts from a good initial configuration, satisfactory results are obtained in practice [37]. In the last iteration SGGA solves the original problem, i.e. minimizes energy function which corresponds to the graph G , but it uses a pre optimized configuration which is obtained by the sequence of previous iterations. It is known that it each iteration of clock ICM the energy is decreasing. Therefore, the convergence of SGGA to local minimum is guaranteed.

4.2 Dynamic programming

4.2.1 Dynamic programming: general idea

The term *dynamic programming* (DP) was originally used in the 1940s by Richard Bellman [38] to describe the process of solving problems where one needs to find the best decisions one after another. In this section we quote the author.

To begin with, the theory of DP was created to treat the mathematical problems arising from the study of various multi – stage decision processes, which may roughly be described in the following way: we have a physical system whose state at any time t is determined by a set of quantities which we call state parameters, or state variables. At certain times, which may be prescribed in advance, or which may be determined by the process itself, we are called upon to make decisions which will affect the state of the system. These decisions are equivalent to transformations of the state variables, the choice of a decision being identical with the choice of a transformation. The outcome of the preceding decisions is to be used to guide the choice of future ones, with the purpose of the whole process that of minimizing (maximizing) some function of the parameters describing the final state.

The basic idea of the theory of DP is that of viewing an optimal policy as one determining the decision required at each time in terms of the current state of the system. Following this line of thought, the basic functional equations given below describing the quantitative aspects of the theory are uniformly obtained from the following intuitive

Principle of optimality. *An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions.*

To illustrate the type of functional equation that arises from an application of the principle of optimality, we consider the simplest case of discrete deterministic process where the system is described at any time by an M – dimensional vector $p = (p_1, p_2, .., p_M)$, constrained to lie within some region D. Let $T = \{T\}_k$, where k runs over a set which may be finite, enumerable, or continuous, be a set of transformations with the property $p \in D$ implies that $T_k(p) \in D$ for all k.

Let us assume that we are considering an N – stage process to be carried out to minimize some scalar function, $R(p)$ of the final state. A policy of a selection of N transformations, $\mathbf{P} = (T_1, T_2, ..., T_N)$, yielding successively the states

$$\begin{aligned} p_1 &= T_1(p), \\ p_2 &= T_2(p_1), \\ &\cdots \\ p_N &= T_N(p_{N-1}). \end{aligned} \tag{4.2}$$

If D is a finite region, if each $T_k(p)$ is continuous in p, and if $R(p)$ is a continuous function of p for $p \in D$, it is clear than an optimal policy exists. The minimum value of $R(p_N)$, determined by an optimal policy, will be a function only of the initial vector p and the number of stages N. Let us then define

$$f_N(p) = \min_p R(p_N) \tag{4.3}$$

the N – stage return obtained using an optimal policy starting from the initial state p .

To derive a functional equation for $f_N(p)$, we employ the principle cited above. Assume that we choose some transformation $T_k(p)$. The minimum return from the

following $(N-1)$ – stages is, by definition, $f_{N-1}(T_k(p))$. It follows that k must now be chosen so as to minimize this. The result is the basic functional equation

$$\begin{aligned} f_N(p) &= \min_k f_{N-1}(T_k(p)) \\ N &= 2, 3, \ldots . \end{aligned} \tag{4.4}$$

It is clear that knowledge of any particular optimal policy, not necessarily unique, will yield $f_N(p)$, which is unique. Conversely, given the sequence $\{f_N(p)\}$, all optimal policies may be determined.

We thus have a duality between the space of functions and the space of policies which is of great theoretical and computational importance.

4.2.2 The DP algorithm on scanlines for graduated graph approximation

Dynamic programming is a powerful and efficient optimization tool which can find global optimal solution of graphs without loops regardless of sub-modularity. Graph-cut based move-making algorithms (e.g. α – expansion and $\alpha\beta$ – swap) are restricted on submodular functions due to the condition of graph-cuts. Even though those algorithms are efficient, we focus on more difficult cases via combining DP and block ICM for efficient move-making on general energy functions.

According to procedure of graduated graph approximation which has been discussed in Chapter 3, edges are gradually added between scanlines over the sequence from Γ. Therefore, each scanline has connections with neighboring scanlines in $\widetilde{G}_t$. To efficiently apply DP, we follow the preprocessing described in the work of [11]. In particular, we update unary potentials of nodes within the current scanline S_s

before optimization. Let the objective function for the current scanline S_s formulate as follows:

$$e_s = \sum_{i \in \mathcal{V}_s} (\psi_i(x_i) + \sum_{j \in N_{out}(i)} \psi_{ij}(x_i, \widetilde{x}_j)) + \sum_{(i,j) \in \mathcal{E}_s} \psi_{ij}(x_i, x_j), \tag{4.5}$$

where $N_{out}(i)$ is a set of fixed neighboring nodes $\widetilde{x}_j$ for node i, which is outside the current scanline. The second sum in the function (4.5) represents pairwise potentials of edges between the current scanline and its neighbors. Actually these potentials are not pairwise because one variable in the pair is fixed. We first modify the unary potentials $\psi_i(x_i)$ within S_s as

$$\widetilde{\psi}_i(x_i) = \psi_i(x_i) + \sum_{j \in N_{out}(i)} \psi_{ij}(x_i, \widetilde{x}_j). \tag{4.6}$$

Replacing with (4.6), the function (4.5) can be rewritten as follows:

$$e_s = \sum_{i \in \mathcal{V}_s} \widetilde{\psi}_i(x_i) + \sum_{(i,j) \in \mathcal{E}_s} \psi_{ij}(x_i, x_j). \tag{4.7}$$

Now DP can find global minimum of the function (4.7) in polynomial time and update current labeling of nodes in S_s. Specifically, the DP algorithm computes the following cumulative cost $C(x_i)$ for each node within S_s and for each possible label in Λ :

$$C(x_i) = \widetilde{\psi}_i(x_i) + \min_{x_{i-1} \in \Lambda} (C(x_{i-1}) + \psi_{i-1,i}(x_{i-1} x_i)). \tag{4.8}$$

First node in S_s is initialized, for which a recursive term in (4.8) is inactive. Then by induction each value $C(x_i)$ is computed by minimizing $C(x_{i-1})$ over Λ for the previous node. Thus, the algorithm computes $|\mathcal{V}_s||\Lambda|$ values for S_s . This procedure is illustrated in Figure 4.1.

The complexity of this dynamic procedure is $O(\|V_s\||\Lambda|^2)$.

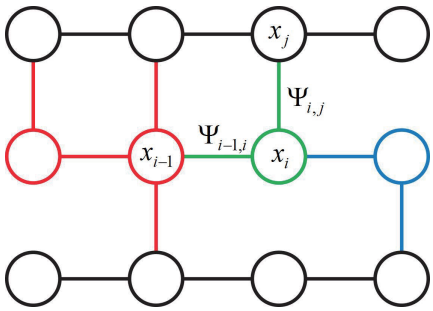


Figure 4.1: DP for the approximated graph $\widetilde{G_t}$. Labels for the gray nodes are fixed. Cumulative costs C for the red nodes have been computed and the algorithm is now processing the green node x_i . To compute the cumulative cost $C(x_i)$, the algorithm minimizes over terms involving possible assignments x_{i-1} to the previous node. Blue nodes will be processed subsequently.

4.2.3 The DP algorithm on trees

DP is a technique to efficiently optimize energy function on chain. We can also apply DP to the nodes of a tree. The idea is to associate a value with each node that combines the values defined for each of its children-in some cases we may process the children in a fixed order. Analogous to regular DP, the message-passing algorithm is applied to optimize intermediate energy functions in the algorithm (2) which has been described in Chapter 3.

The message-passing formulation of the DP algorithm can be generalized to find marginal distributions over individual variables and the MAP estimate in a tree structured graphical model. The resulting algorithm is known as belief propagation [4] and has two variants: max-product, for computing the MAP solution, and sum-product, which allows computation of marginals of individual random variables.

Max-product message passing is similar in spirit to DP algorithms. Like the

Viterbi [39] algorithm, it works by passing messages between tree nodes in two stages. In the first stage, messages are passed from the leaf nodes to their parents, which in turn pass messages to their parents, and so on until the messages reach the root node. The message $m_{i\to j}$ from a node i to its parent j is computed as

$$m_{i\to j}(x_j) = \max_{x_i} P(x_j, x_i) \prod_{k\in N_c(i)} m_{k\to i}(x_i), \tag{4.9}$$

where $N_c(i)$ is the set of all children of node i. The MAP label of the variable at the root r of the tree can be computed as

$$\hat{x}_r = \operatorname*{argmax}_{x_r} \prod_{k\in N_c(r)} m_{k\to r}(x_r). \tag{4.10}$$

Given the MAP label $\hat{x}_p$ of a variable $\mathbf{x}$, the label of any of its children i can be found as

$$\hat{x}_i = \max_{x_i} P(\hat{x}_p, x_i) \prod_{k\in N_c(i)} m_{k\to i}(x_i). \tag{4.11}$$

Chapter 5

Experiments

5.1 Introduction

In this section, we evaluate efficiency of our SGGA algorithm on a synthetic experiment and OpenGM2 benchmark dataset. All the experiments have been performed on Intel i5-2500 3.3GHz CPU and 8GB RAM.

5.2 Image deconvolution

Firstly we test our algorithm on a synthetic image deconvolution problem. Image deconvolution is a task of restoring a blurry and noisy image [40]. Due to the highly connected structure and strong non – submodularity, this problem has been reported as a challenging one [41]. In particular, the difficult problem degrades the performance of graph cut – based algorithms. We prepare the same MRF models as in [40] for comparison.

The goal of image deconvolution is to recover an image X from its convolution with a known blurring function h. This is equivalent to solving the linear inverse

problem:

$$Y = HX, \tag{5.1}$$

for X given Y, where H is the convolution matrix corresponding to h. In this work we consider the generalized image deconvolution problem, where H is an arbitrary nonnegative matrix. Inverse problems of this form are ill – posed, and are typically solved by minimizing a regularized energy function. The energy of a solution X is given by

$$\|Y - HX\|_2^2 + G(X), \tag{5.2}$$

which is the sum of a unary term and a pairwise term.

A natural class of pairwise terms is

$$G_{MRF}(X) = \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j). \tag{5.3}$$

The neighborhood $\mathcal{E}$ consists of pairs of adjacent pixels, usually the 4 – connected neighbors. The pairwise term $\psi(l, l')$ gives the cost to assign l and l' to neighboring pixels. Typically the pairwise term has a form such as $\psi(l, l') = \min(|l - l'|, K)$ for some metric $|\cdot|$ and constant K.

The problem we address is to efficiently minimize

$$E(X) = \|Y - HX\|_2^2 + G_{MRF}. \tag{5.4}$$

When H is diagonal, as in image denoising, the unary term has a restricted form that makes it computationally tractable to minimize E. Specifically,

$$\|Y - HX\|_2^2 = \sum_i (y_i - H_{i,i}x_i)^2, \tag{5.5}$$

which means that the unary term for the pixel i to have the hypothesized label (i.e. intensity) x_i only depends on x_i and y_i. With such an H, the energy E can be efficiently minimized by graph cuts.

Graph cuts, however, can only be applied to certain energy functions, i.e. binary submodular. Existing graph cut energy minimization methods cannot be applied when H is non – diagonal. Intuitively, this is because the unary term for a pixel to have a label depends on the hypothesized labels of other nearby pixels.

In this work we consider the generalized deconvolution problem with non – diagonal matrix H. Consider the correlation matrix of H defined by $R_H(i,j) = \sum_{i=1}^{N} H_{r,i}H_{r,j}$. We can then write

$$\|Y - HX\|_2^2 = \sum_i y_i^2 - 2\sum_i (\sum_j y_j H_{j,i})x_i + \sum_i R_H^2(i,i)x_i x_j. \tag{5.6}$$

For experiments images (colored with three labels) are blurred with 3×3 Gaussian kernel with $\sigma = 3$. Then the image is distorted with Gaussian noise with $\sigma = 10$. The pairwise potentials are imposed with Potts model for smoothness prior. We have tested all the comparing algorithms on three datasets: "Characters" dataset (5 images), "White chessmen" dataset (6 images) and "Black chessmen" dataset (6 images). Some of images from these datasets are shown in Figure 5.1. Examples of original images and distorted ones are shown in Figures 5.2, 5.3, 5.4.

We compare SGGA with conventional deterministic algorithms: alpha-expansion, TRW-S and Belief Propagation (BP) and with MCMC – GD, a hybrid algorithm recently proposed by Kim and Lee [12]. A source code of the last algorithm has been

provided by the authors.

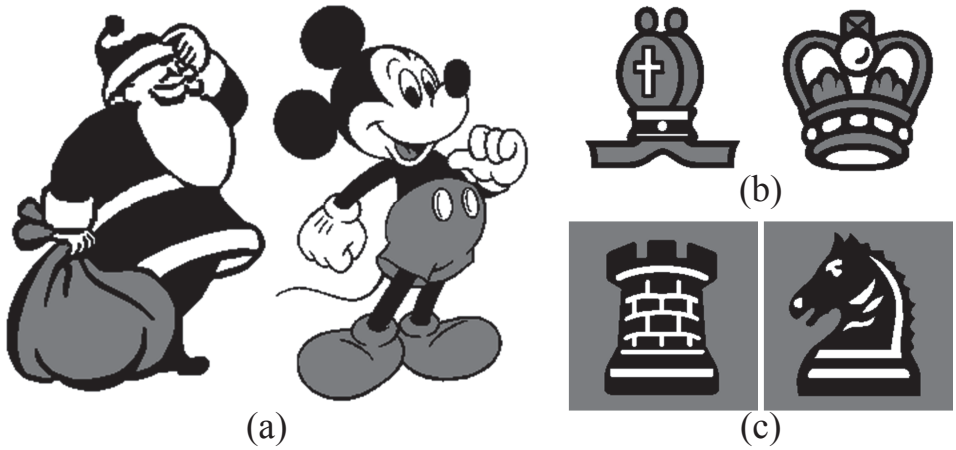


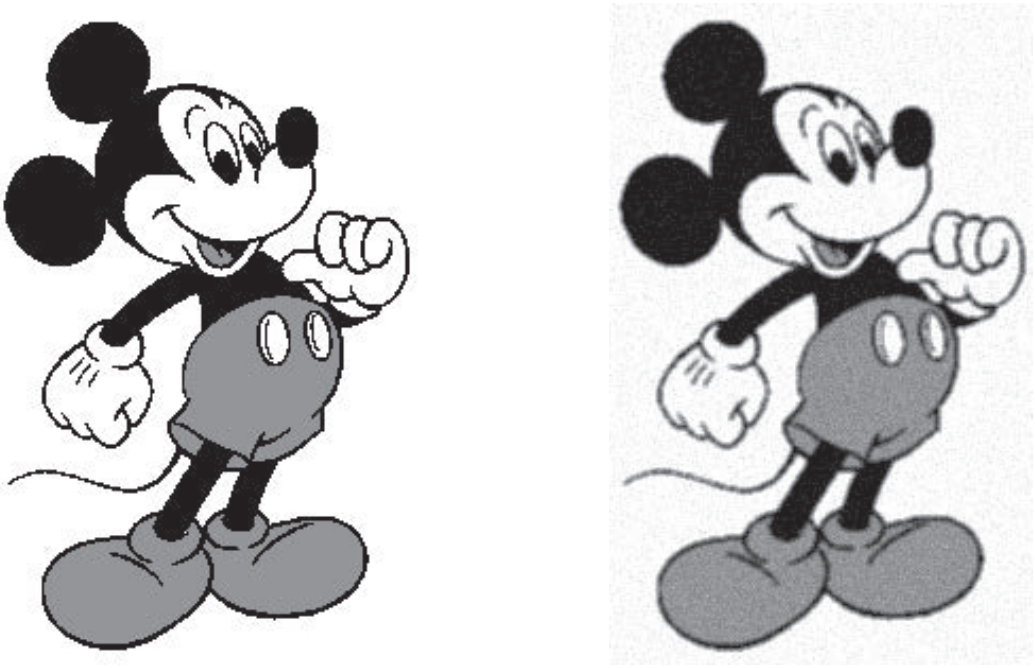
Figure 5.1: Representative images from datasets for image deconvolution problem

The optimal schedule of adding edges for our algorithm has been found by cross-validation: $\rho = 0.9$. For MCMC – GD algorithm, an optimal relative rate of a dynamic and static anchor-based proposal is set as $Q_D = 0.1$.

Results of experiments for image deconvolution problem are summarized in Table 5.1.

Our algorithm achieves the lowest energy over all the test images as well as the lowest average error rate. It converges to a local minimum faster than MCMC – GD on average. Deterministic algorithms achieve much higher energy than SGGA but converge faster. The fastest algorithm among deterministic ones is α – expansion, which converges in 3.51 seconds on average. Overall the deterministic algorithms are efficient in running time but converge to higher energy state than the others. Our algorithm provides the lowest values with comparable running time to the deterministic methods.

Visual result of experiments with the image "Mickey" from the "Characters"



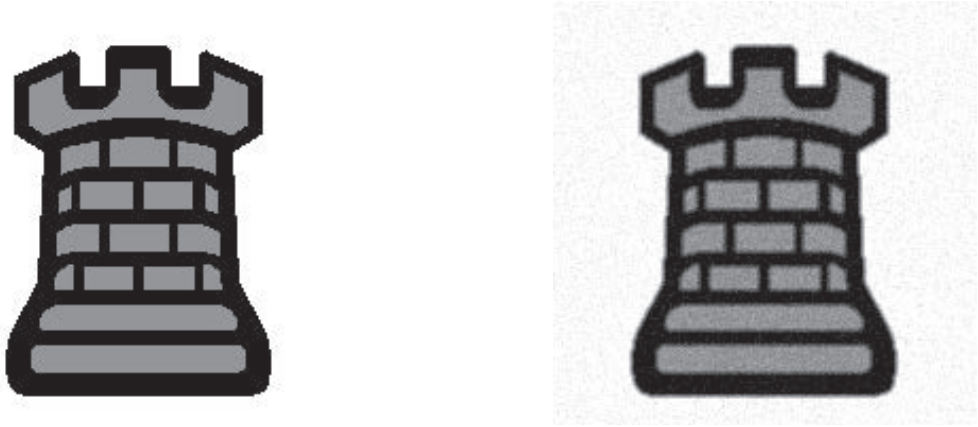
(a) Original image (b) Corrupted image

Figure 5.2: An example from the dataset "Characters", the image "Mickey": (a) the original image , (b) the image after blurring and distortion. The image is blurred with 3×3 Gaussian kernel with $\sigma = 3$. Then the image is distorted with Gaussian noise with $\sigma = 10$.

dataset, "White rook" from the "White chessmen" dataset and "Black king" from the "Black chessmen" dataset are shown in Figures 5.5, 5.6 and 5.7. The convergence of energy states of these experiments are shown in Figures 5.8, 5.9, 5.10.

We also test our algorithm on more challenging problem. Images are blurred with 3×3 Gaussian kernel with $\sigma = 7$. Then the image is distorted with Gaussian noise with $\sigma = 100$ (see Figure 5.12).

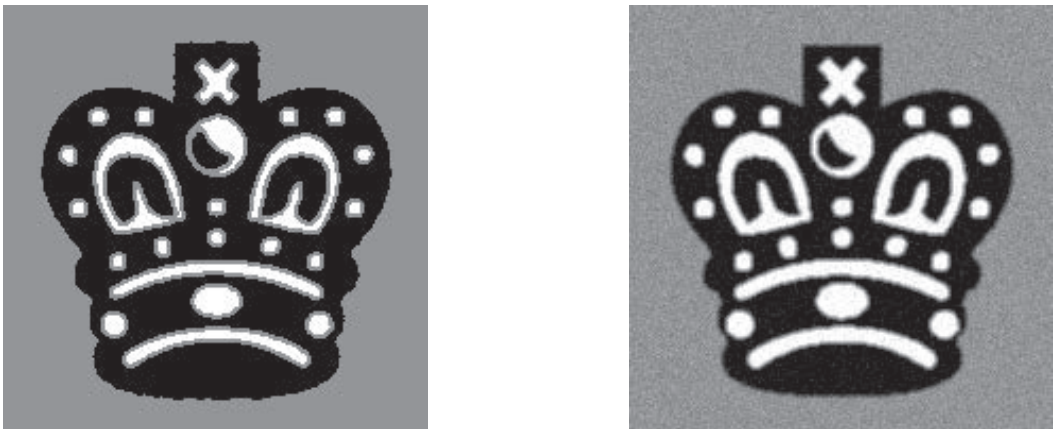
In Figure 5.11 graphs of energies for the image "Santa" obtained by the com-



(a) Original image

(b) Corrupted image

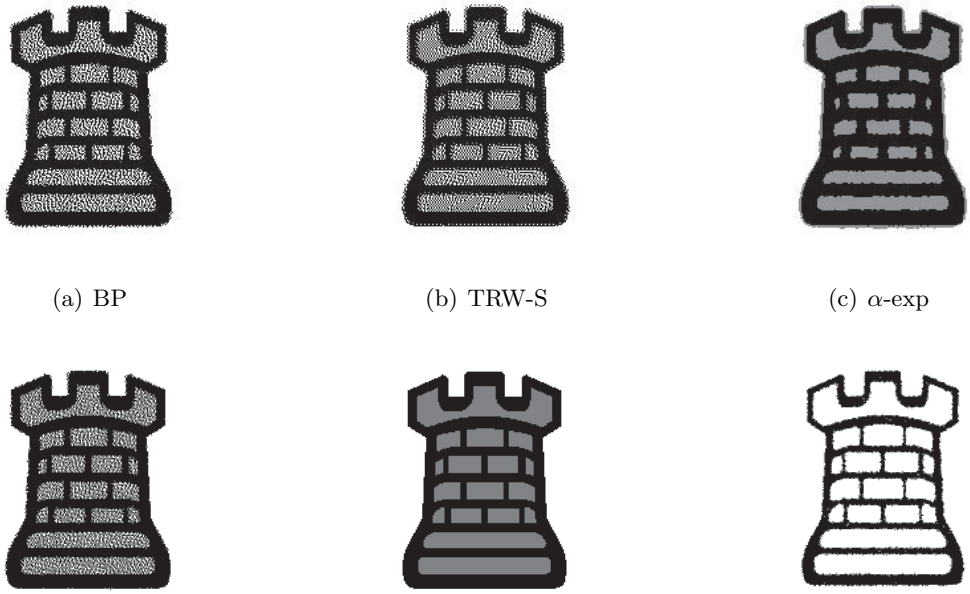
Figure 5.3: An example from the dataset "Characters", the image "White rook": (a) the original image , (b) the image after blurring and distortion. The image is blurred with 3×3 Gaussian kernel with $\sigma = 3$. Then the image is distorted with Gaussian noise with $\sigma = 10$.



(a) Original image

(b) Corrupted image

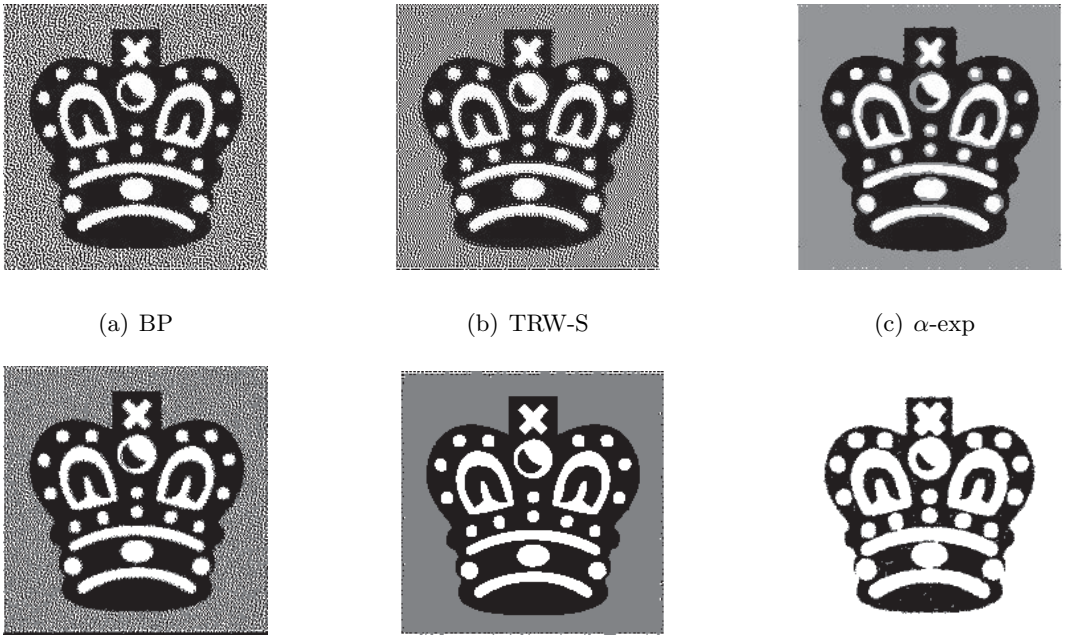
Figure 5.4: An example from the dataset "Characters", the image "Black king": (a) the original image , (b) the image after blurring and distortion. The image is blurred with 3×3 Gaussian kernel with $\sigma = 3$. Then the image is distorted with Gaussian noise with $\sigma = 10$.



(a) BP (b) TRW-S (c) α-exp

(d) MCMC-GD (e) SGGA (f) STGA

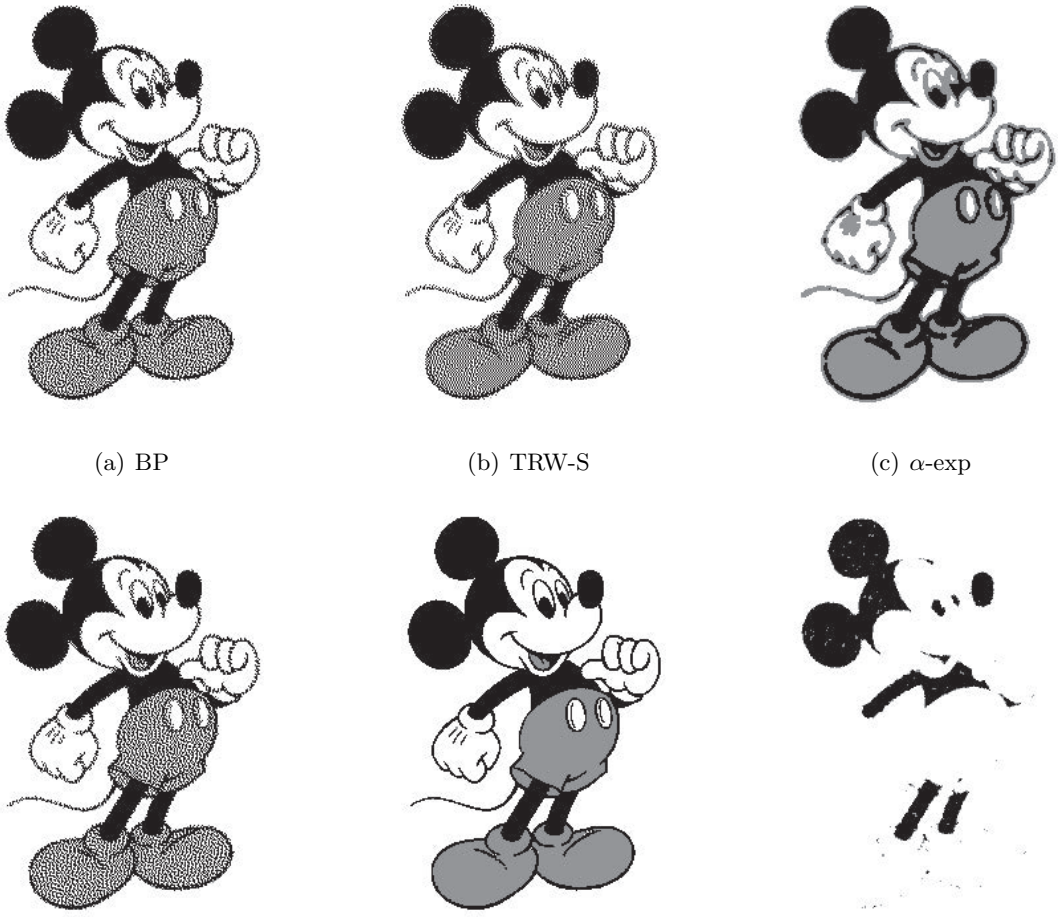
Figure 5.5: Visual results of experiments with image "White rook" - from "White chessmen". Results obtained by each algorithm are shown.



(a) BP (b) TRW-S (c) α-exp

(d) MCMC-GD (e) SGGA (f) STGA

Figure 5.6: Visual results of experiments with image "Black king" - from "Black chessmen". Results obtained by each algorithm are shown.



(a) BP (b) TRW-S (c) α-exp

(d) MCMC-GD (e) SGGA (f) STGA

Figure 5.7: Visual results of experiments with images "Mickey" from the dataset "Characters". Results obtained by each algorithm are shown.

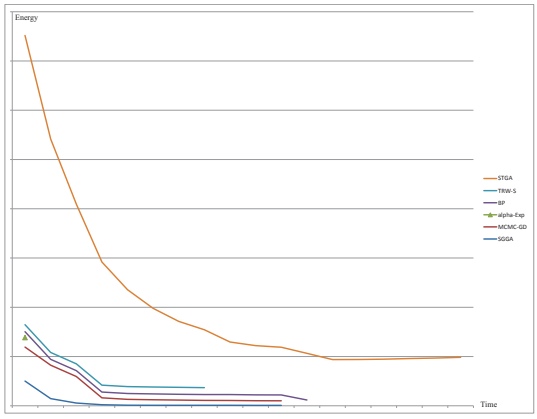


Figure 5.8: Numerical results of experiments with the image "White rook". The proposed algorithm SGGA achieves the lowest energy and converges faster than stochastic MCMC-GD.

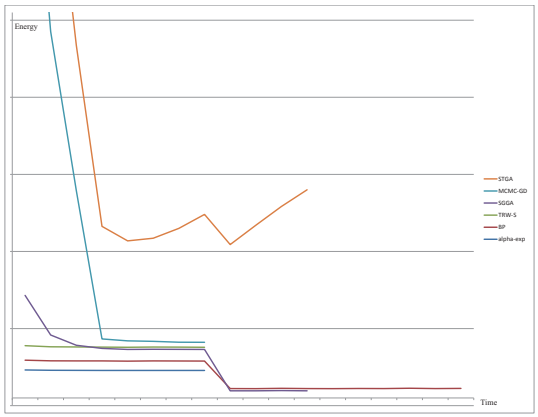


Figure 5.9: Numerical results of experiments with the image "Black king". The proposed algorithm SGGA achieves the lowest energy and converges faster than stochastic MCMC-GD.

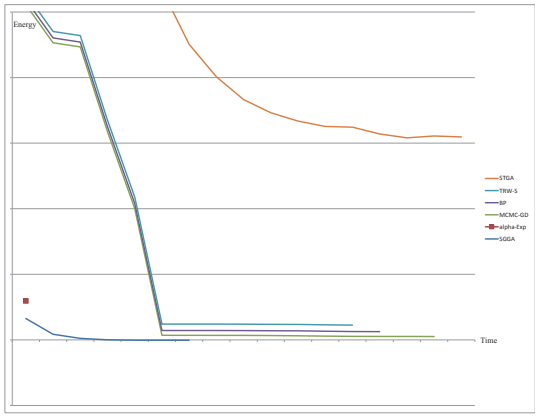


Figure 5.10: Numerical results of experiments with the image "Mickey". The proposed algorithm SGGA achieves the lowest energy and converges faster than stochastic MCMC-GD.

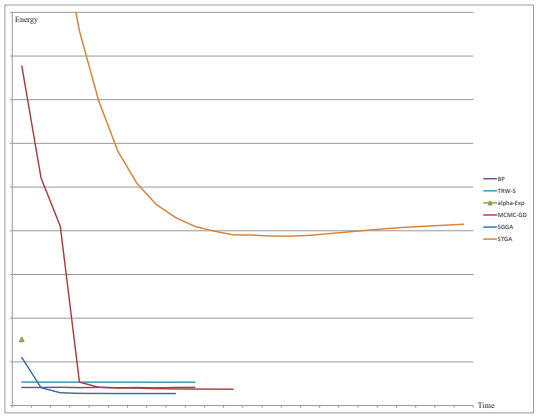
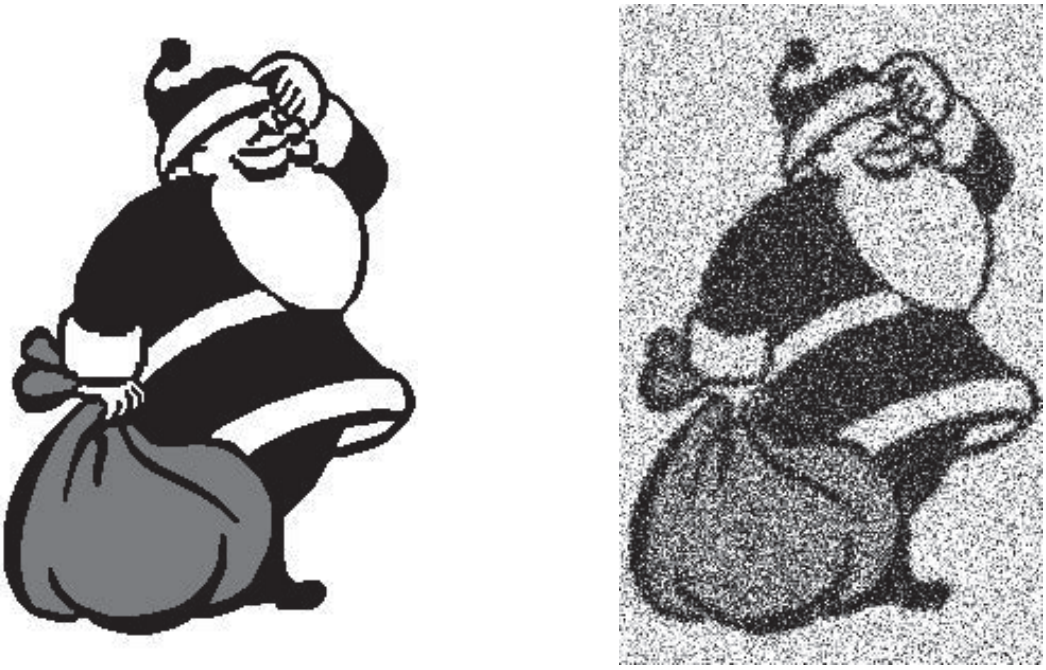


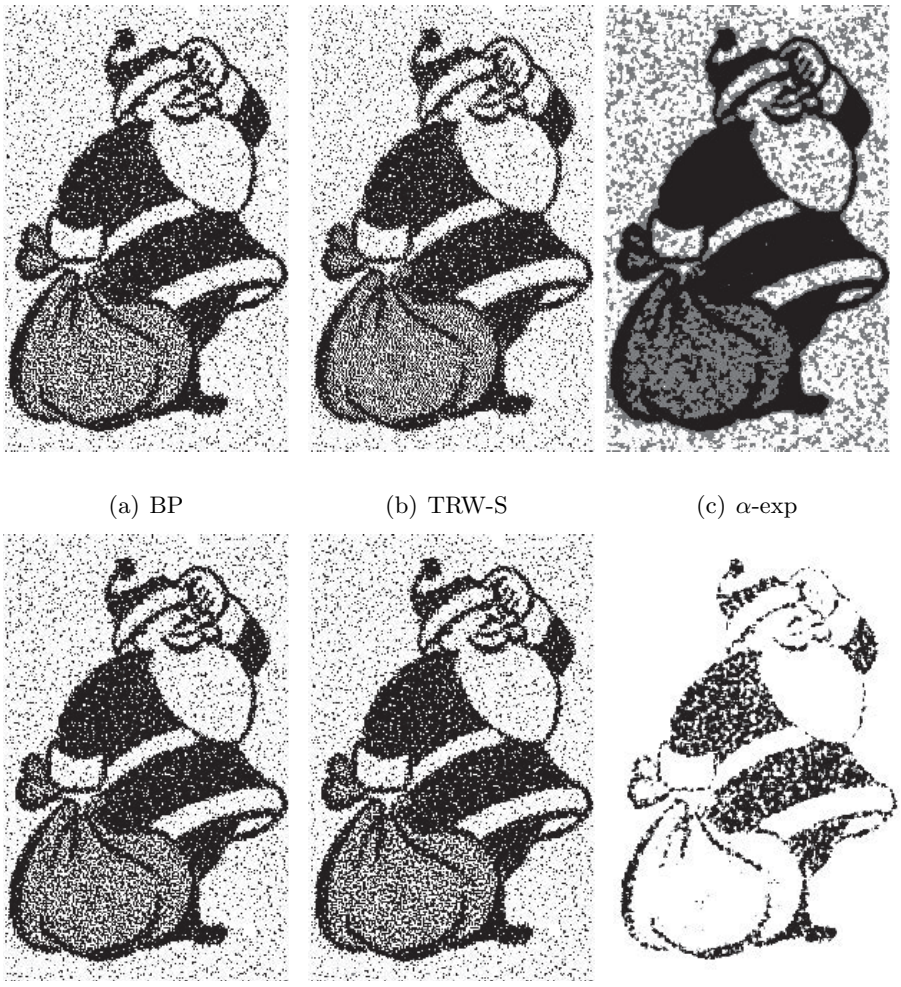
Figure 5.11: Numerical results of experiments with the image "Santa". More challenging problem is considered. The image is blurred with 3×3 Gaussian kernel with $\sigma = 7$. Then the image is distorted with Gaussian noise with $\sigma = 100$. The proposed algorithm SGGA achieves the lowest energy and converges faster than stochastic MCMC-GD.



(a) Original image

(b) Corrupted image

Figure 5.12: An example from the dataset "Characters", the image "Santa" (a) the original image , (b) the image after blurring and distortion. The image is blurred with 3×3 Gaussian kernel with $\sigma = 7$. Then the image is distorted with Gaussian noise with $\sigma = 100$.



(a) BP (b) TRW-S (c) α-exp

(d) MCMC-GD (e) SGGA (f) STGA

Figure 5.13: Visual results of experiments with image "Santa" from the dataset "Characters". Results obtained by each algorithm are shown.

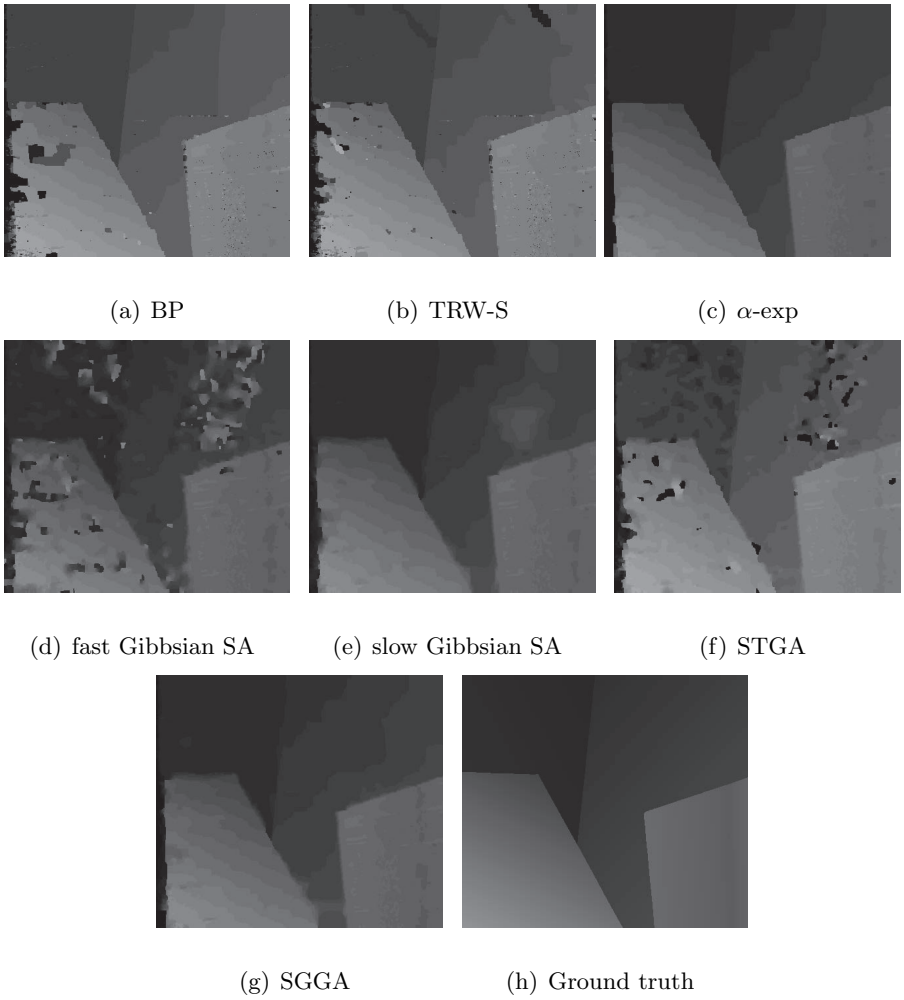
pared algorithms are shown. In Figure 5.13 corresponding visual results are depicted. Energy values and running time for the compared algorithms are reported in Table 5.2.

5.3 OpenGM2 benchmark

We also evaluate the proposed method on OpenGM2 benchmark [2]. This benchmark includes various types of energy functions in different applications. We compare the performance of SGGA with conventional stochastic algorithm Gibbsian SA, and also with conventional deterministic algorithms. We report average energies of SGGA,STGA and Gibbsian SA. SGGA achieves the lowest energy in the Stereo problem (see Table 5.3). The images obtained by all comparable algorithms are shown in Figure 5.14.

Furthermore, for color segmentation problem, inpainting-n4, and image restoration SGGA gives competitive results compared to deterministic algorithms. Obtained images for these problems are reported in Figure 5.15, 5.16 and 5.17. Energies and running time are reported in Tables 5.4, 5.5 and 5.6, respectively. Gibbsian SA algorithm gives the worst results compared to SGGA and the deterministic algorithms and produces visually poor images.

Scheduling the portion of added edges is an important control parameter in our algorithm. According to our experiments, the parameter ρ should be high enough. In Figure 5.18, we report energies obtained by SGGA for different values of ρ.It justifies our intuition that energy function should be approximated slightly from one to another. With a low value of ρ the algorithm initially solves a complex problem and gets stuck in a "bad" local minimum because it has smaller range of possible

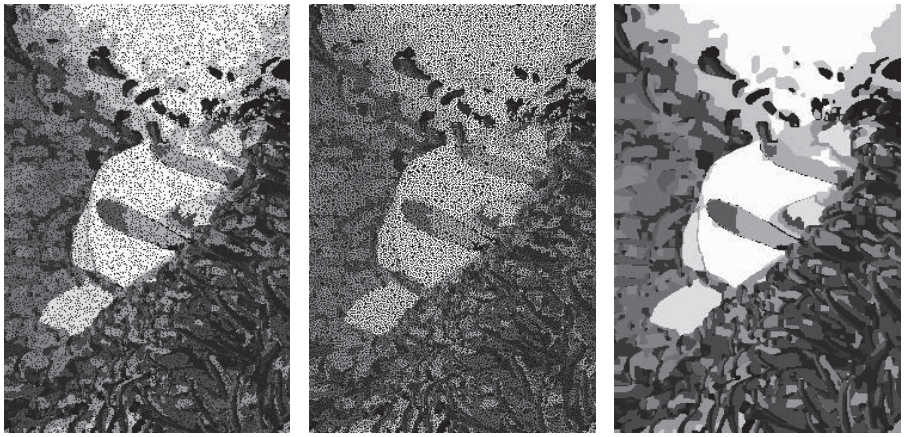


(a) BP (b) TRW-S (c) α-exp

(d) fast Gibbsian SA (e) slow Gibbsian SA (f) STGA

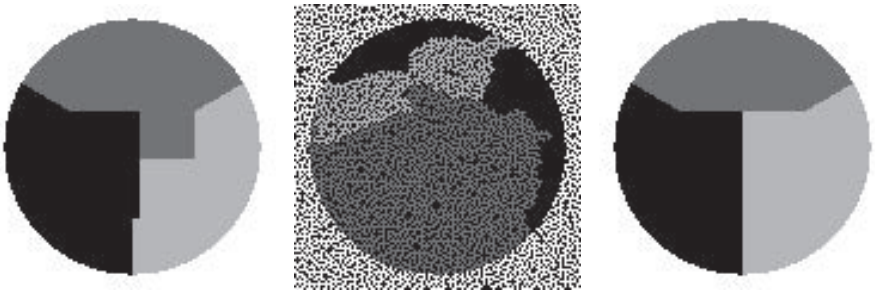
(g) SGGA (h) Ground truth

Figure 5.14: Results of experiments with Stereo problem, image "Venus". Results obtained by each algorithm are shown.



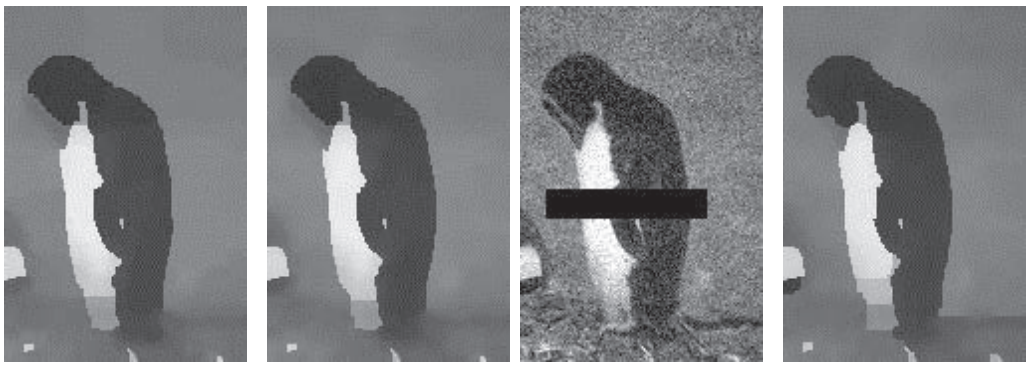
(a) Gibbsian SA (b) STGA (c) SGGA

Figure 5.15: Color segmentation: visual results of experiments with the image "Clown-fish" of Gibbsian SA, STGA and SGGA respectively. SGGA gives a bit worse result than deterministic algorithms (difference with TRW-S=3.4 %). SGGA achieves visually similar results to deterministic methods.

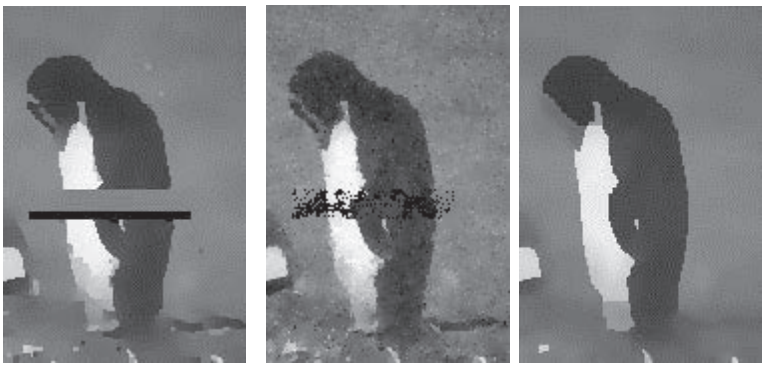


(a) Gibbsian SA (b) STGA (c) SGGA

Figure 5.16: Inpainting-n4: visual results of experiments. SGGA gives the same result as deterministic algorithms. Images obtained by Gibbsian SA, STGA and SGGA are reported. STGA gives poor result.



(a) BP (b) TRW-S (c) Input (d) α-exp



(e) Gibbsian SA (f) STGA (g) SGGA

Figure 5.17: Results of experiments with restoration problem, image "Penguin". TRW-S and SGGA algorithms show best results.

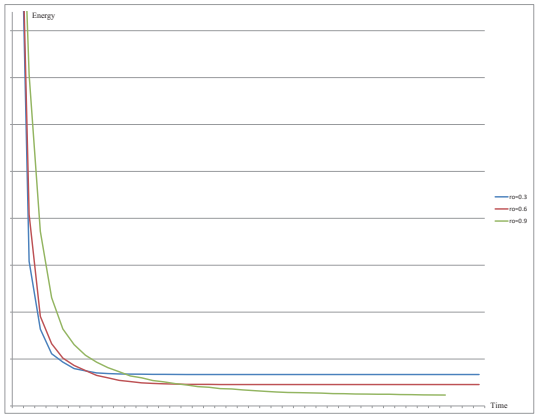


Figure 5.18: Stereo: the image "Venus". Energies obtained by SGGA with different schedules of adding edges.

moves.

Table 5.1: Image deconvolution results. Energies, average error rates and average convergence time are reported. Mean values of energies for SGGA and MCMC-GD are reported.

	Energy ($\times 10^6$)					
	SGGA	MCMC-GD	α-Exp	BP	TRW-S	STGA
[characters dataset]						
Santa	**-3.72**	5.57	15.78	8.84	14.40	767.469
Pororo	**-3.35**	5.40	35.24	8.45	15.72	329.005
Mickey	**-2.90**	8.57	35.29	12.51	18.26	352.52
Rodin	**-4.75**	1.66	27.72	4.04	9.67	239.98
Gangnam	**1.17**	2.79	20.23	34.84	37.97	655.941
[white chessmen dataset]						
white king	**-0.08**	5.15	16.94	6.80	10.26	203.139
white queen	**-1.45**	7.43	16.11	9.40	12.48	245.658e
white rook	**-0.50**	5.24	17.18	7.16	9.20	214.694
white bishop	**-0.62**	5.01	12.96	6.83	9.24	193.108
white knight	**-0.56**	5.77	13.61	8.09	9.99	219.863
white pawn	**-0.93**	3.12	8.45	4.10	5.39	132.122
[black chessmen dataset]						
black king	**1.87**	19.24	14.12	23.53	27.88	446.691
black queen	**1.61**	20.67	14.13	24.19	26.88	464.596
black rook	**1.59**	20.76	13.66	25.78	28.15	474.631
black bishop	**1.36**	22.74	13.60	27.21	29.66	505.85
black knight	**1.41**	20.35	12.78	24.16	26.56	451.905
black pawn	**0.97**	23.10	7.25	28.89	30.20	518.15
Average Error	**1.22** %	6.2%	8.18%	34.47%	35.98%	65.98%
Average running time,sec	5.46	7.33	**3.51**	6.43	3.82	9.91

Table 5.2: Energy values and running time obtained by the compared algorithms for deconvolution of the image "Santa".

Methods	BP	TRW-S	α-exp	MCMC-GD	STGA	SGGA
Energy ($\times 10^8$)	5.41049	5.53433	6.51233	5.37346	9.27569	**5.2756**
Time, sec.	10.139	10.12	**0.14**	23.901	13.288	1.76

Table 5.3: Energy values and running time obtained by the compared algorithms for stereo matching of the image "Venus".

Methods	BP	TRW-S	α-exp	Fast Gibbsian SA	Slow Gibbsian SA	STGA	SGGA
Energy ($\times 10^6$)	4.456640	4.276870	3.13305	96.9633	3.17854	4.03015	**3.10828**
Time, sec.	23.47	22.59	**14.25**	300.175	300.175	75.544	70.461

Table 5.4: Energy values and running time obtained by the compared algorithms for color segmentation of the image "Clown – fish".

Methods	BP	TRW-S	α-exp	Gibbsian SA	STGA	SGGA
Energy	14820.1	**14817.5**	14826.5	27234	42125.9	14867.5
Time, sec.	10.139	10.12	**0.14**	30.11	38.99	1.33

Table 5.5: Energy values and running time obtained by the compared algorithms for inpainting – n4 problem.

Methods	BP	TRW-S	α-exp	Gibbsian SA	STGA	SGGA
Energy	**484.591**	**484.591**	**484.591**	502.655	227303	**484.591**
Time, sec.	0.03	0.19	**0.02**	24.63	33.19	0.38

Table 5.6: Energy values and running time obtained by the compared algorithms for image restoration problem.

Methods	BP	TRW-S	α-exp	Gibbsian SA	STGA	SGGA
Energy ($\times 10^7$)	1.77758	**1.66137**	1.73497	2.00826	8.54049	**1.66203**
Time, sec.	29.167	**24.029**	28.318	34.951	48.97	31.44

Chapter 6

Conclusion

In this work, we have proposed a new efficient algorithm for solving discrete multi-label MRF optimization problems. Our algorithm is based on the concept of graduated optimization. The algorithm initially minimizes a simplified function and progressively proposes a set of simpler functions that the original function. In the procedure, we have contributed how to generate the sequence of intermediate functions, which is not well known in discrete MRF problems. Our algorithm combines advantages of stochastic and deterministic methods. We have exploited a stochastic method to generate intermediate functions, which guides the intermediate solutions to the near-optimal solution for the original problem. The transition from one intermediate problem to another is controlled by the schedule of gradual addition of edges. Then we have utilized efficient deterministic algorithm - block ICM and dynamic programming. However, we have controlled only the amount of active edges, which allows various range of move-making. We have shown that our algorithm generally achieves lower energy than conventional deterministic and stochastic optimization algorithms in solving image deconvolution problem and convergence rate similar to

deterministic algorithms. Also the efficiency of our SGGA has been supported by the set of experiments with problems from the OpenGM2 benchmark.

Bibliography

[1] S. Bagon and M. Galun, "A unified multiscale framework for discrete energy minimization," *CoRR*, vol. 4967, 2012.

[2] J. H. Kappes, B. Andres, F. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. Kausler, X. Bernhard, J. Lellmann, N. Komodakis, and C. Rother, "A comparative study of modern inference techniques for discrete energy minimization problem," *The IEEE Conference on Computer Vision Pattern Recognition*, 2013.

[3] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[4] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," *The Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 900–907, 2003.

[5] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.

[6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[7] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 65–81, 2004.

[8] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[9] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H.Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[10] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[11] W. Kim and K. M. Lee, "Scanline sampler without detailed balance: An efficient mcmc for mrf optimization," *The IEEE Conference on Computer Vision Pattern Recognition*, pp. 1354–1361, 2014.

[12] W. Kim and K. M. Lee, "A hybrid approach for mrf optimizations problems: Combination of stochastic sampling and deterministic algorithms," *Computer Vision and Image Understanding*, vol. 115, pp. 1623–1637, 2011.

[13] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA, USA: MIT Press, 1987.

[14] A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 1048–1054, 1989.

[15] D. Terzopoulos, "The computation of visible-surface representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 417–438, 1988.

[16] A. Witkin, D. Terzopoulos, and M. Kass, "Signal matching through scale space," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 133–144, 1987.

[17] A. Boccuto, M. Discepoli, I. Gerace, R. Pandolfi, and P. Pucci, "A gnc algorithm for deblurring images with interacting discontinuities," *Proc. VI SIMAI*, pp. 296–310, 2002.

[18] S. Tirthapura, D. Sharvit, P.Klein, and B. Kimia, "Indexing based on edit-distance matching of shape graphs," in *Photonics East (ISAM, VVDC, IEMB)*. International Society for Optics and Photonics, 1998, pp. 25–36.

[19] T. Brox, "From pixels to regions: partial differential equations in image analysis," Ph.D. dissertation, Faculty of Mathematics and Computer Science, Saarland University, Germany, April 2005.

[20] A. Rangarajan and R. Chellappa, "Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation," in *IEEE 10th International Conference on Pattern Recognition*, vol. 2, 1990, pp. 127–133.

[21] R. M. Dufour, E. L. Miller, and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *IEEE Transactions on Image Processing*, vol. 11, no. 12, pp. 1385–1396, 2002.

[22] A. Leich, M. Junghans, and H.-J. Jentschel, *Hough transform with GNC*. NA, 2004.

[23] J. Zerubia and R. Chellappa, "Mean field annealing using compound gauss-markov random fields for edge detection and image estimation," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 703–709, 1993.

[24] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[25] B. L.Price, B. Morse, and S. Cohen, "Simultaneous foreground, background, and alpha estimation for image matting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2157–2164.

[26] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *Journal of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[27] M. Isard and A. Blake, "Condensation-conditional density propgation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[28] Y. Suh, M. Cho, and K. M. Lee, "Graph matching via sequential monte carlo," *European Conference on Computer Vision. Springer Berlin Heidelberg*, pp. 642–637, 2012.

[29] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[30] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," *The IEEE Conference on Computer Vision Pattern Recognition*, vol. 2, pp. 1075–1082, 2005.

[31] C. Zach, M. Sormann, and K. Karner, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," *The Third IEEE International Symposium on Processing, Visualization and Transmission*, pp. 512–518, 2006.

[32] O.Veksler, "Stereo correspondence by dynamic programming on a tree," *The IEEE Conference on Computer Vision Pattern Recognition*, pp. 384–390, 2005.

[33] S. Hermann, "Evaluation of scanline optimization for 3d medical image registration," *The IEEE Conference on Computer Vision Pattern Recognition*, pp. 3073–3080, 2014.

[34] M. J. Wainwright, T. S. Jaakola, and A. S. Willsky, "Map estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches," *The IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 3697–3717, 2005.

[35] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.

[36] D. Batra, A. Gallagher, D. Parikh, and T. Chen, "Beyond trees: Mrf inference via outer-planar decomposition," *The IEEE Conference on Computer Vision Pattern Recognition*, pp. 2496–2503, 2010.

[37] S.Theodoridis and R. Chellappa, *Academic Press Library in Signal Processing.* Kidlington,Oxford, UK: Academic Press of Elsevier, 2014.

[38] R. Bellman, *The theory of dynamic programming.* The RAND CORP. Santa Monica, CA, 1954.

[39] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[40] A. Raj and R. Zabih, "A graph-cut algorithm for generalized image deconvolution," *The IEEE International Conference on Computer Vision*, pp. 1048–1054, 2005.

[41] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," *The IEEE Conference on Computer Vision Pattern Recognition*, pp. 1–8, 2007.

국문 초록

마르코프 랜덤 필드 (MRF) 모델은 컴퓨터 비전 분야에서 매우 중요한 모델이다. 하지만 대부분의 기존 알고리즘들은 주어진 에너지 함수에 대하여 정확한 해를 효율적으로 찾지 않고, 계산 효율을 위하여 최적 해의 근사치만을 구한다는 한계가 있다. 본 학위 논문에서는 일반적인 다중 라벨 MRF 최적화 문제를 풀기 위해 순차적 최적화 기법을 적용한 확률론적 순차적 그래프 근사 (SGGA) 알고리즘을 제시한다. 제시된 알고리즘은 먼저 간소화된 함수의 최적 해를 구한 뒤, 원래의 목적 함수와 같아질 때까지 순차적으로 함수를 변환한다. 그러나, 중간 과정의 함수들을 정해주는 것과 각 함수들이 변환될 때 사용하는 매개 변수들의 종류와 값을 정해주는 것은 매우 어려운 문제이다. 이를 해결하기 위해 본래의 목적 함수를 순차적으로 근사하는, 새로운 반복적 방법을 제안한다. 중간 과정의 함수들을 생성하기 위하여 확률론적 방법이 사용되고, 이 방법은 중간 과정의 최적 해들을 본래의 목적 함수에 대한 최적 해와 거의 유사한 값을 갖도록 유도해 준다. 중간 함수들 사이의 변환은 순차적으로 에지를 추가해 줌으로써 조절된다. 매 반복마다 블록 ICM과 같은 결정론적 알고리즘이 적용되어 중간 함수들의 해를 찾고 다음 중간 함수의 초기 해를 생성하는 방식이다. 제안된 알고리즘은 지역 최적해로의 수렴이 보장되어 있다. 제안된 알고리즘의 성능을 평가하기 위하여 합성된 영상 디콘볼루션 문제와, OpenGM2 벤치마크의 다양한 문제에 대하여 실험이 시행되었다.

주요어: : MRF, discrete optimization, graph approximation.

학번: 2013-22506