Ph.D. DISSERTATION

Hybrid Approaches for MRF Optimization: Combination of Stochastic and Deterministic Methods

MRF 최적화를 위한 새로운 접근:
확률론적 방법과 결정론적 방법의 결합

BY

WONSIK KIM

February 2014

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

Hybrid Approaches for MRF Optimization: Combination of Stochastic and Deterministic Methods

MRF 최적화를 위한 새로운 접근:
확률론적 방법과 결정론적 방법의 결합

지도교수 이 경 무

이 논문을 공학박사 학위논문으로 제출함

2014 년 2 월

서울대학교 대학원

전기컴퓨터공학부

김 원 식

김원식의 공학박사 학위논문을 인준함

2014 년 2 월

위 원 장 : 李商郁 Sang-Uk, Lee

부위원장 : 李炅武

위 원 : 李相旭

위 원 : 朴仁圭

위 원 : 林鍾宇

Abstract

Markov Random Field (MRF) models are of fundamental importance in computer vision. Many vision problems have been successfully formulated in MRF optimization. They include stereo matching, segmentation, denoising, and inpainting, to mention just a few. To solve them effectively, numerous algorithms have been developed. Although many of them produce good results for relatively easy problems, they are still unsatisfactory when it comes to more difficult MRF problems such as non-submodular energy functions, strongly coupled MRFs, and high-order clique potentials.

In this dissertation, several optimization methods are proposed. The main idea of proposed methods is to combine stochastic and deterministic optimization methods. Stochastic methods encourage more exploration in the solution space. On the other hand, deterministic methods enable more efficient exploitation. By combining those two approaches, it is able to obtain better solution. To this end, two stochastic methodologies are exploited for the framework of combination: Markov chain Monte Carlo (MCMC) and stochastic approximation.

First methodology is the MCMC. Based on MCMC framework, population based MCMC (Pop-MCMC), MCMC with General Deterministic algorithms (MCMC-GD), and fusion move driven MCMC (MCMC-F) are proposed. Although MCMC provides an elegant framework of which global convergence is provable, it has the slow convergence rate. To overcome, population-based framework and combination with deterministic methods are used. It thereby enables global moves by exchanging information between samples, which in turn, leads to faster mixing rate. In the view of optimization, it means that we can reach a lower energy state rapidly.

Second methodology is the stochastic approximation. In stochastic approximation, the objective function for optimization is approximated in stochastic way. To apply this approach to MRF optimization, graph approximation scheme is proposed for the approximation of the energy function. By using this scheme, it alleviates the problem of non-submodularity and partial labeling. This stochastic approach framework is combined with graph cuts which is very efficient algorithm for easy MRF optimizations. By this combination, fusion with graph approximation-based proposals (GA-fusion) is developed.

Extensive experiments support that the proposed algorithms are effective across different classes of energy functions. The proposed algorithms are applied in many different computer vision applications including stereo matching, photo montage, inpaining, image deconvolution, and texture restoration. Those algorithms are further analyzed on synthetic MRF problems while varying the difficulties of the problems as well as the parameters for each algorithm.

Key words: Markov random fields, Combinatorial optimization, Markov chain Monte Carlo, Population based algorithm, Stochastic approximation, Non-submodular energy model, Higher order energy model

Student number: 2007-20950

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Markov random field

1.1.1 MRF and Gibbs distribution

Markov Random Field (MRF) models are of fundamental importance in computer vision. Many vision problems have been successfully formulated in MRF optimization. They include stereo matching, segmentation, denoising and inpainting, to mention just a few. Recently, Szeliski *et al.* [2] presented a comprehensive review of the standard MRF-based vision problems and the comparative results of existing optimization algorithms.

The general formulation of the MRF models is as follows. Let $\mathbf{X} = \{X_1, \cdots, X_N\}$ be a set of random variable. Each random variable X_i takes a value x_i. The set of random variable $\mathbf{X}$ is said to be an MRF if and only if it satisfies the following Markovian property:

$$p(x_i|x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_N) = p(x_i|\mathbf{x}_{\mathcal{N}_i}), \tag{1.1}$$

where $\mathcal{N}_i$ is a set of index of neighbors of ith random variable and

$$\mathbf{x}_{\mathcal{N}_i} = x_k | k \in \mathcal{N}_i. \tag{1.2}$$

An MRF is often represented by a graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each random variable X_i corresponds to ith node and the neighboring system $\mathcal{N}_i$ is defined by the set of edges $\mathcal{E}$.

The Hammersley-Clifford theorem established that the joint probability of any MRF can be represented by Gibbs distribution. It is so called Markov-Gibbs equivalence. The Gibbs distribution of an MRF $\mathbf{X}$, defined on the graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$ with the neighboring system $\mathcal{E}$, is given by

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{\rfloor} \phi_{\rfloor}(\mathbf{x}_{\rfloor}), \tag{1.3}$$

where Z is a normalizing constant and $\phi_{\rfloor}$ is a clique potential function defined on the set of random variable $\mathbf{x}_{\rfloor}$ for the clique $\rfloor$, that is,

$$\mathbf{x}_{\rfloor} = x_k | k \in \rfloor. \tag{1.4}$$

1.1.2 MAP estimation and energy minimization

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

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}). \tag{1.5}$$

Often, the energy function is considered instead the joint probability function because of the simplicity and the computational issue. The energy function defined on an MRF is given by

$$E(\mathbf{x}) = -\ln p(\mathbf{x}) + Z \tag{1.6}$$

$$E(\mathbf{x}) = \sum_{\rfloor} \theta_{\rfloor}(\mathbf{x}_{\rfloor}), \tag{1.7}$$

where

$$\theta_{\rfloor}(\cdot) = -\ln \phi_{\rfloor}(\cdot). \tag{1.8}$$

Now the MAP solution $\mathbf{x}^*$ can be represented by the energy function as following:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} E(\mathbf{x}). \tag{1.9}$$

Therefore, estimating the MAP solution for the given MRF is equivalent to finding the solution $\mathbf{x}$ which minimizes the energy function (1.7).

1.1.3 MRF formulation for computer vision problems

To formulate energy function, it is often more convenient to express the energy function as sum of the several terms according to clique size, That is,

$$E(\mathbf{x}) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \alpha \sum_{\langle s,t \rangle \in \mathcal{E}} \theta_{st}(x_s, x_t) + \beta \sum_{\langle \in \mathcal{H}} \theta_{\langle}(\mathbf{x}_{\langle}), \tag{1.10}$$

where α and β are the weight factors, $\mathcal{H}$ is the set of higher-order cliques, and $\mathbf{x}_{\langle}$ is the set of random variable which corresponds to the vertices in the clique $\langle \in \mathcal{H}$.

The first term $\theta_s(x_s)$ is called the unary term or data term and is defined in various ways depending on the applications. For example, in stereo problem it can be intensity difference, sum of squared difference or Birchfield-Tomasi measure of corresponding pixels. In denoising problem, it can be the intensity difference between the true and the noisy pixels. In the segmentation problem, it can be the color difference between a single pixel and the histogram of the segment it belongs to.

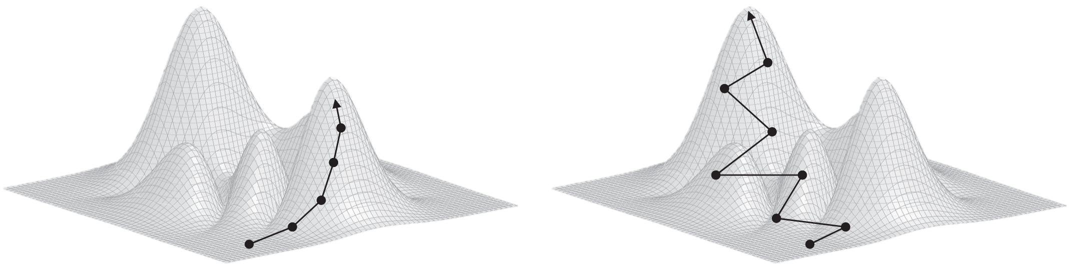
The second term $\theta_{st}(x_s, x_t)$ is called the pairwise term or smoothness term. This term usually encodes the prior knowledge into the energy function. In most applications, smoothness regularization constraints are commonly used, which compel the solution to be piecewise smooth. Widely-used smoothness models include the Potts model, the truncated linear model and the truncated quadratic model. The MRF formulation in computer vision has long been limited up to pairwise terms due to the weak minimizing power of the optimization methods. The third term $\theta_{\langle}(\mathbf{x}_{\langle})$, which is called higher-order term, has been introduced to overcome the limitations of the energy models with only up to pairwise terms. The higher-order term encodes more complex and realistic knowledge about the scene [3, 4, 5].

Recently, there has been increasing emphasis on the higher-order term [5, 6, 4, 7, 8, 9, 10, 11]. This term is introduced to design more sophisticated probability model. They are usually modeled by capturing more information from the statistics of images. Although this term helps to obtain much better solutions, it often makes the problems intractable.

1.2 Optimizing energy function

Many algorithms have been proposed to solve MRF problems. The existing algorithms can be divided into two approaches: deterministic and stochastic sampling algorithms. Some of the well-known deterministic algorithms are move-making algorithms. Move-making algorithms iteratively make local moves to explore the solution space. They include Iterated Conditional Modes (ICM), the Gradient Descent Algorithm and Graph Cuts [12, 13, 14]. Graph Cuts are the state of the art among those move-making algorithms. It becomes more powerful due to recent advances including the fusion move and the Quadratic Pseudo-Boolean Optimization (QPBO) algorithm [15, 5]. Graph Cuts iteratively optimize the binary sub-problems of the original problem. They are fast, accurate and even find global optima for some classes of functions. Another important class of deterministic algorithms is the message passing approach. It includes Belief Propagation (BP) [16] and Tree Reweighted Message Passing (TRW) [17, 18]. BP was originally developed for graphs without cycles. Although there is no guarantee of convergence in the case of the graph with cycles, it has been successfully applied to vision problems. One of the important properties of TRW is that it gives a lower bound of the energy function, which can be used to check optimality of the solutions.

Although those methods have been successively applied to various problems, the story becomes different when it comes to more difficult MRF problems. There are some known factors which make MRF problems more difficult: non-submodular functions, strongly coupled MRF models, high connectivity and higher-order clique potentials. It is known that more non-submodular terms make the problem harder [15]. The coupling strength also affects performance in solving MRF problems. The coupling strength refers to the relative strength of pairwise versus unary terms. As coupling strength increases, problems become more difficult [2, 19]. High connectivity of graphs is another factor which makes the problem difficult [20]. Higher-order clique potential also make the problem difficult. Despite difficulty, higher-order clique potential has often been used to improve the results in some vision applications [6, 5].



(a) deterministic

(b) stochastic

Figure 1.1: The comparison between (a) the deterministic and (b) the stochastic optimization methods. Since the deterministic methods always move to the higher probability state in greedy way, it is easily stuck at local optima. On the other hand, the stochastic methods allow more exploration over the solution space. In consequence, it leads the solution to the global optima.

More difficult MRF models are inevitable to incorporate realistic image priors into the models. (*e.g.*, occlusion terms in stereo and texture information in denoising and segmentation.) To those difficult examples, most existing algorithms are not applicable, and even with some applicable algorithms the results are far from the global optimum.

To overcome the limitations of previous approaches, stochastic optimization methods are considered in this dissertation. Stochastic optimization refers a set of methods which obtain the solution in probabilistic way. Stochastic optimization allows the solution to explore over the solution space more than deterministic ones. Figure 1.1 compares deterministic and stochastic optimization methods. In deterministic method, the solution is updated always to the higher-probability state. On the contrary, stochastic method randomly updates the solution to the higher- or lower-probability states. Consequently, stochastic optimization is able to avoid getting stuck at local minima and achieves better results than deterministic one. In this dissertation, two different methods are considered to develop new algorithms. They are Markov chain Monte Carlo and stochastic approximation.

1.2.1 Markov chain Monte Carlo

MCMC algorithms have been used to sample from the target distribution $p(\mathbf{x})$. It generates a sequence of samples $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots$ using a Markov chain. A tth sample $\mathbf{x}^{(t)}$ is drawn from a conditional distribution $q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})$. We call $q(\cdot|\cdot)$ the kernel of the Markov chain. A kernel $q(\cdot|\cdot)$ is reversible if and only if

$$p(\mathbf{x}^{(t-1)})q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = p(\mathbf{x}^{(t)})q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}). \tag{1.11}$$

This is also called detailed balance condition. If a kernel $q(\cdot|\cdot)$ satisfies detailed balance, the Markov chain process $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots$ generated by the kernel converges to the target distribution $p(\mathbf{x})$.

Along with simulated annealing, MCMC has also been used to obtain an optimum sample of the target function, *i.e.* a sample $\mathbf{x}$ which maximizes $p(\mathbf{x})$. In MCMC, a new sample is drawn from the previous sample with a local transition probability, based on the Markov chain.

Although simulated annealing is proven to converge to optimal solution, it still takes very long time to reach the global optimum becuase most MCMC methods allow only local moves in a large solution space. To overcome the limitations of MCMC methods as an optimizer, recently Swendsen-Wang Cuts (SWC) was proposed [21, 22]. In SWC, it is shown that bigger local moves are possible than in previous methods while maintaining the detailed balance. SWC uses Simulated Annealing (SA) [23] to find the global optimum. Although SWC allows bigger local moves, a very slow annealing process is still needed to approach the global optimum with probability 1. Therefore, we need a faster annealing process for real vision applications. However, fast annealing does not always guarantee the global optimum and the samples are often trapped in local optima.

In this dissertation, population-Based framework [24, 25] is used to overcome the drawbacks of previous MCMC methods. Also, new idea is proposed to exploit the advantages of deterministic algorithms in the framework of MCMC.

1.2.2 Stochastic approximation

Stochastic approximation algorithms are a set of methods which optimize an objective function $f(\mathbf{x})$, which cannot be directly calculated, but only estimated via some approximations. A famous example of stochastic approximation is the stochastic gradient method.

Stochastic gradient method deals with an objective function that has the form of a sum:

$$f(\mathbf{x}) = \sum_i f_i(\mathbf{x}). \tag{1.12}$$

For each iteration, this method updates the solution by the following equations.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla f_i(\mathbf{x}), \tag{1.13}$$

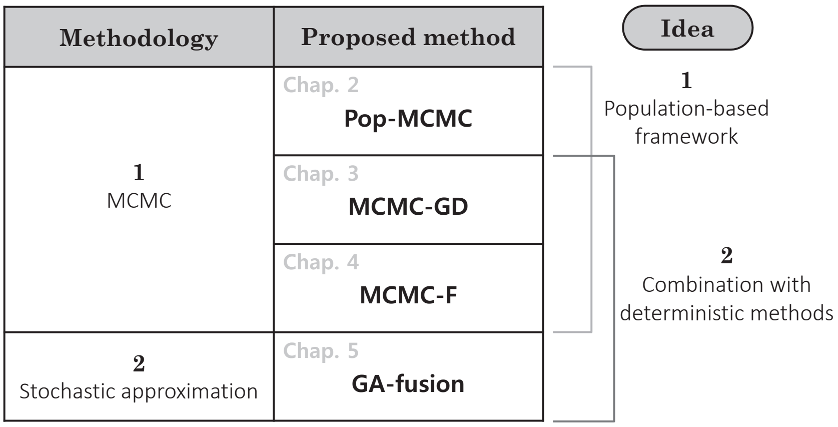
where η is a step size. This process is similar to that of gradient descent method except the fact that it calculates the approximated gradient only using a single instance of $f_i(\mathbf{x})$. Sometimes, a subset of $f_i(\mathbf{x})$ is used instead of a single instance. This method is often used for training parameters with large data set. The aim for training is to find parameters $\mathbf{x}$, which minimize loss function $f(\mathbf{x})$. However, calculation of the gradient $\nabla f(\mathbf{x})$ is often computationally expensive. In this case, we can consider stochastic gradient method using the update scheme (1.13). When the objective function is convex or pseudoconvex, appropriate scheduling of the step size η leads solution to the global minimum [26].

1.3 combination of stochastic and deterministic methods

One of the main ideas in this dissertation is to combine stochastic and deterministic methods to deal with difficult MRF problems. There are two interpretations why combination approach achieves better performance. On the one hand, the stochastic methods are boosted by combination with deterministic algorithms. Stochastic methods alone generally cannot achieve large and effective move. By the help of deterministic methods, a stochastic method achieves more efficient exploration in the solution space. Moreover, deterministic methods guide it to make effective jumps from one basin to another over the energy barrier. Consequently, this property yields faster convergence and better solutions. On the other hand, the stochastic methods help deterministic algorithms not to be stuck in local minima. Every deterministic method ends up in one of the local minima. With the help of stochastic methods, however, it can escape from the local minima and keep searching for better solutions.

1.4 Outline of dissertation

The structure of the dissertation and the main ideas are summarized in Figure 1.2. The main goal of this dissertation is to combine stochastic and deterministic methods



Methodology	Proposed method
1 MCMC	Chap. 2 **Pop-MCMC**
	Chap. 3 **MCMC-GD**
	Chap. 4 **MCMC-F**
2 Stochastic approximation	Chap. 5 **GA-fusion**

Idea
1
Population-based framework
2
Combination with deterministic methods

Figure 1.2: Summary of the methods proposed in this dissertation. Two methodologies are used to develop stochastic optimization algorithms. They are Markov chain Monte Carlo (MCMC) and stochastic approximation. Also, two main ideas are used, which are population-based framework and combination with existing deterministic methods.

to achieve better performance. Two different framework of stochastic optimization are considered. In Chapter 2–4, MCMC based methods introduced. Chapter 5 proposes a new method based on stochastic approximation. To combine deterministic methods within the framework of MCMC, population-based approach is exploited. Chapter 2 proposes Pop-MCMC algorithm [27, 28] designed for MRF optimization. On this population-based framework, several new methods are proposed by combination with deterministic methods (Chapter 3–4). Chapter 3 proposes MCMC-GD algorithm [29, 30]. In MCMC-GD, an elegant approach for combining deterministic methods within the framework of Pop-MCMC are developed. This combination strategy is extended to solve continuous optimization problems in Chapter 4, in which MCMC-F algorithm [31] is introduced. To allow more active exploration in the solution space, other stochastic optimization framework called stochastic approximation is considered. By combining graph cuts (deterministic method) and stochastic approximation, GA-fusion algorithm [32] is developed.

Chapter 2

Population-based Markov Chain Monte Carlo

2.1 Introduction

Markov random field (MRF) have been used in numerous areas in computer vision [2]. MRFs are generally formulated as follows. Given a graph $G = (\mathcal{V}, \mathcal{E})$, the joint probability function of the pairwise MRF is given by

$$p(\mathbf{x}) \propto \prod_{i \in \mathcal{V}} \phi_i(x_i) \cdot \prod_{(i,j) \in \mathcal{E}} \phi_{i,j}(x_i, x_j), \tag{2.1}$$

where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $x_i \in \{1, 2, \cdots, L\}$ is the label assigned on node i. Obtaining maximum a posteriori (MAP) solution of probability (2.1) is NP-hard in general cases. To achieve better approximation solutions, many different optimization methods have been developed.

Graph cuts-based methods are fast and provide very low energy solution with standard 4-neighborhood benchmark problems. However, it can be applied to a limited class of energy functions [14]. [2] showed that α-expansion move method was faster and slightly better than $\alpha\beta$-swap move method in all cases in their experiments. However, α-expansion move method can be applied to more limited class of energy functions than $\alpha\beta$-swap move. BP (Belief Propagation) is a message passing method originally developed for graphs without cycles. In general, although it is not guaranteed to converge, it has been successfully applied to loopy graphs [33]. TRW (Tree-reweighted message passing) is also a message passing method [17]. It finds lower energy solution than Graph cuts in many problems. An important property of TRW is that it gives a lower bound on the energy which can be used to check how close our solution to the global minimum energy. All of the deterministic methods are approximation algorithms. Although Graph cuts provides global minimum for some restricted energy models, none of these methods guarantee to obtain the global minimum solution for a general stereo model in practical time since it is known to be an NP hard problem.

In contrast to the deterministic methods, stochastic approaches such as sampling-based methods can be used to find global optimum. Sampling-based methods were originally developed to generate samples from a given target distribution or to integrate functions in high dimensional space. These Sampling-based methods are also have been used for statistical estimation and optimization. In this chapter, we use a sampling-based method for energy minimization to solve the stereo matching problem.

The Monte-Carlo method is the most primitive sampling-based method. In this method, a new sample is drawn depending on a pre-determined proposal distribution. This distribution is independent on previous samples. However, there are some difficulties in applying the Monte Carlo methods to vision problems as an optimizer.

In general, we need to solve vision problems in very high-dimensional solution spaces. Even if it is assumed to be 100 pixels in the width and height, respectively, the dimension of the image space can be as high as 10^4. Monte Carlo methods would take infinitely long time since the acceptance rate would be almost zero in such a high-dimensional case. Moreover, we need to design a proper proposal distribution close to the target distribution. To resolve these problems, Markov Chain Monte Carlo (MCMC) methods had been tried. In MCMC, a new sample is drawn from the previous sample with a local transition probability, based on the Markov chain. Contrary to simple Monte Carlo methods, the acceptance rates of MCMC methods are high enough, and the proposal distributions are designable even in high-dimensional problems. Therefore, MCMC methods are more appropriate for the application to vision problems than the Monte Carlo methods. However, difficulties still remain in applying MCMC to vision problem as an optimizer. Since most MCMC methods allow only local moves in a large solution space, it still takes very long time to reach the global optimum.

To overcome the limitations of MCMC methods as an optimizer, recently Swendsen-Wang Cuts (SWC) was proposed [21, 22]. In SWC, it is shown that bigger local moves are possible than in previous methods while maintaining the detailed balance. SWC uses Simulated Annealing (SA) [23] to find the global optimum. Although SWC allows bigger local moves, a very slow annealing process is needed to approach the global optimum with probability 1. This is an apparent drawback of SWC. Therefore, we need a faster annealing process for real vision applications. However, fast annealing does not always guarantee the global optimum and the samples are often trapped in local optima.

In this chapter, we propose a new MCMC method called Population-Based

MCMC (Pop-MCMC) [24, 25] that can overcome the drawbacks of SWC. Our goal is to obtain the lower energy state faster than other sampling methods including SWC which have been previously applied to this problem. In Pop-MCMC, two or more samples are drawn at the same time. Samples can exchange information with each other. This makes it possible to perform global moves of samples. It means that the mixing rate of drawn samples becomes faster. And in the view of optimization, the faster mixing rate means that it takes shorter time for the samples to approach the global optimum than conventional methods.

This chapter presents the design of Pop-MCMC for MRF optimization. The proposed algorithm is applied to stereo matching and compared with previous methods. The chapter is organized as follows: In Section 2.2, SWC and Pop-MCMC are briefly introduced. Then, we present how Pop-MCMC is applied to Gibbs distribution on MRF model in Section 2.3. Section 2.4 gives the experimental results. In the final Section 2.5, we summarize the chapter with discussions.

2.2 Related Works

In this section, we first describe the SWC, which has been applied to vision problems [21, 22]. Then, we present the description of Pop-MCMC.

2.2.1 Swendsen-Wang Cuts

Swendsen-Wang Cuts (SWC) originated from the Swendsen-Wang (SW) method. Swendsen and Wang proposed SW method in 1987 [34]. It overcame the slow convergence of previous sampling-based methods such as Gibbs sampler. Let us explain SW briefly.

We consider a 2-D lattice graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges connecting neighboring nodes. Each node $v \in \mathcal{V}$ is assigned a label $x_i \in \{1, 2, \cdots, L\}$. The number of possible labels is L. In a 2-D lattice graph, each node has four edges. We assume that this graph follows the Potts model, which is often used in vision as a prior model. The formulation of Potts model is as follows.

$$p(\mathbf{x}) = \frac{1}{Z} \exp \beta \sum_{\langle i,j \rangle \in \mathcal{E}} \mathbf{1}(x_i = x_j), \tag{2.2}$$

where $\mathbf{x}$ represents $(x_1, \cdots, x_N)$ and N is the number of nodes. Z and β are constants. $\mathbf{1}(\cdot)$ represents a Boolean function. When the graph follows the Potts model, a global minimum should be the states in which all the nodes have the same labels.

In Gibbs sampler, the label of only one node can be flipped to generate the next sample. So it needs a generation of $O(L^N)$ samples to reach the global optimum. In contrast, in SW the labels of a cluster of nodes are flipped at the same time.

However SW has several drawbacks. It assumes fixed number of labels, and does not create new labels in the case when the number of labels is unknown. And it is only applicable to Ising/Potts model. In addition, it does not consider the external field, such as the observed visual data in vision.

To overcome the above limitations of SW, SWC has been proposed by extending SW from the Metropolis-Hastings perspective [21, 22]. SWC can be applicable to arbitrary posterior probabilities, and can incorporate external data easily. The summary of SWC is described in the following:

Assume that a current state is A, repeat the process below.

1. If the labels of two neighboring nodes s and t are different, the edge connecting two nodes is removed. If the labels are the same, we determine whether the

edge is retained or not with the probability q_e. If there exists external field, we consider it in designing the probability q_e. This process is repeated for all edge $e = \langle s, t \rangle \in \mathcal{E}$. Then nodes connected by remaining edges are considered as a cluster.

2. One cluster $\mathcal{V}_0$ is randomly selected.

3. New label l' of the chosen cluster $\mathcal{V}_0$ is proposed with a proposal distribution $q(l'|\mathcal{V}_0, A)$.

4. Determine whether we accept the newly generated sample (or state B) with acceptance probability α by the following Metropolis-Hastings rule.

$$\alpha = \min\left(1, \frac{q(\mathcal{V}_0 \mid B)q(l \mid \mathcal{V}_0, B)p(B \mid I)}{q(\mathcal{V}_0 \mid A)q(l' \mid \mathcal{V}_0, A)p(A \mid I)}\right), \tag{2.3}$$

where I represents the external field, that is, the observed input image. No matter how q_e and the proposal distribution $q(l'|\mathcal{V}_0, A)$ are designed, the detailed balance is maintained by Metropolis-Hastings kernel. Therefore, we can appropriately design q_e and the proposal distribution of the new label freely, so as to use the information of input image properly.

In order to reduce the complexity of SWC, a modified clustering method, SWC-2 has been proposed [21]. In SWC-2, a connected node cluster $\mathcal{V}_0$ is determined by a recursive method as described in the following.

1. Select a seed node $v \in \mathcal{V}$ randomly, and assign it to a cluster $\mathcal{V}_0$.

2. Repeat until no more node is added to $\mathcal{V}_0$.

   For any edge $e =< s, t >\in \mathcal{E}$ between the node $s \in \mathcal{V}_0$ and its neighboring node $t \notin \mathcal{V}_0$,

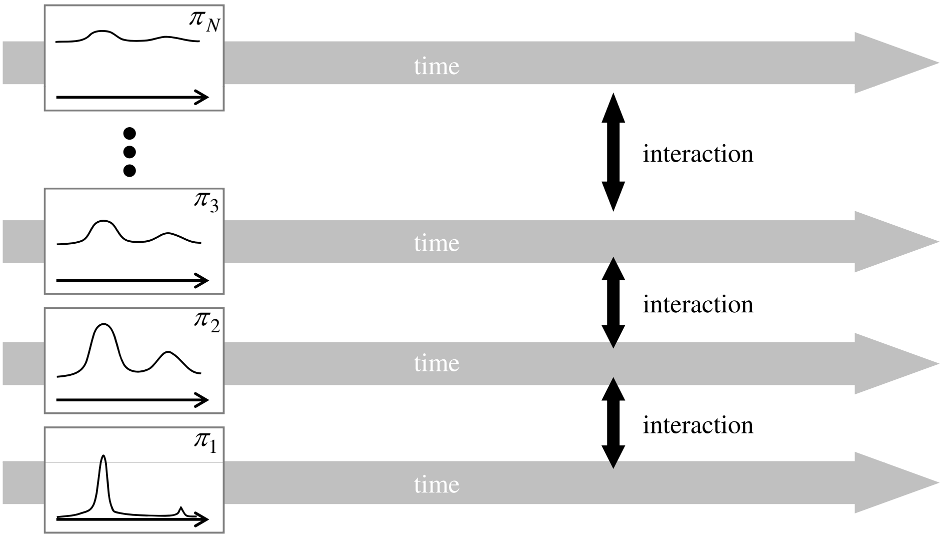


Figure 2.1: Chains in parallel tempering.

(a) If the labels of the two nodes s and t are deferent, remove the edge. Otherwise, determine whether the edge e should be retained or not with probability q_e, same as in SWC.

(b) If the edge e is not removed, add the node t to the cluster $\mathcal{V}_0$.

Note that in constructing $\mathcal{V}_0$, we need to calculate q_e only for the edges at the border of the cluster $\mathcal{V}_0$. It leads to saving of computational costs. In our work, we adopt SWC-2 as a part of the proposed algorithm.

2.2.2 Population-based MCMC

Population-based MCMC (Pop-MCMC) or evolutionary Monte Carlo is a stochastic simulation method that combines a population of Metropolis-Hastings samplers and Evolutionary Algorithm to improve the performance of MCMC samplers. Pop-MCMC generates multiple chains in parallel with different temperatures, and exchanges information among them to accelerate the mixing rate. This method can be considered as a variant of the Parallel Tempering (PT), that was proposed by Geyer [35] in 1991 and modified by others later [36]. PT aims to overcome the problems of traditional single process MCMC using a Metropolis-Hastings update, which has low mixing rate. The basic idea of PT is to simulate multiple replicas of the original system in parallel at a series of different temperatures, and swap the configurations with a Metropolis-Hastings criterion. The target distribution of ith chain is defined as follows.

$$p_i(\mathbf{x}) = p(\mathbf{x})^{\frac{1}{T_i}}, \tag{2.4}$$

where $p(\mathbf{x})$ is an original target distribution, and T_i is the temperature of the ith chain. In the chain with high temperature, the target distribution is nearly flat as depicted in Figure 2.1, where the heights of barriers between local optima are very low. Therefore, the samples in such chain can freely wander in contrast to the samples in a chain with low temperature. By exchanging these higher-temperature configurations with the configuration of a low temperature of our interest, we can allow the low temperature simulation to sample configurations much more efficiently than with local Metropolis updates only. This leads to a faster mixing rate between samples, and helps to escape from local minima.

Pop-MCMC allows chains to exchange information more actively than PT by introducing a new move called the *crossover* move. It originated from the genetic algorithm and then modified to fit the MCMC framework [24]. In Pop-MCMC, the Markov chain state is augmented as the population of all chains. Given an original target distribution $p(\mathbf{x})$, a new expanded target distribution is defined as follows.

$$p^*(\mathbf{x}_{1:N}) = \prod_{i=1}^{N} p_i(\mathbf{x}_i), \tag{2.5}$$

where N is the number of chains to use. We assume that $p_k \equiv p$ for at least one chain $k \in \{1, \ldots, N\}$. $\mathbf{x}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ is a population composed of samples of N chains. Each component $\mathbf{x}_i$ in the vector $\mathbf{x}_{1:N}$ is called as a *chromosome*. The term chromosome is borrowed from genetic algorithm. The goal of Pop-MCMC is to generate samples $\mathbf{x}_{1:N}$ which follow the new target distribution p^*. And a collection of chromosomes from the kth chain, which has the target distribution $p_k = p$, is what we want to obtain finally.

Pop-MCMC has three different types of moves; mutation, exchange and crossover, which are described below in detail.

1. Mutation move

The mutation move updates a chromosome of a single chain using a Markov kernel, while other chains keep unchanged. We can use a conventional MCMC algorithm. Let us suppose that the current population is $\mathbf{x}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N\}$. Among N chains, we randomly select ith chain and generate a new chromosome $\mathbf{y}_i$ from the current chromosome $\mathbf{x}_i$ by an MCMC algorithm. Then, a new population $\mathbf{y}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{y}_i, \cdots, \mathbf{x}_N\}$ is proposed, and is accepted according to the Metropolis-Hastings rule with probability

$$\alpha = \min(1, \gamma_m), \tag{2.6}$$

where

$$\begin{aligned}\gamma_m &= \frac{p^*(\mathbf{y}_{1:N})}{p^*(\mathbf{x}_{1:N})} \cdot \frac{T(\mathbf{y}_{1:N} \to \mathbf{x}_{1:N})}{T(\mathbf{x}_{1:N} \to \mathbf{y}_{1:N})} \\ &= \frac{p_i(\mathbf{y}_i)}{p_i(\mathbf{x}_i)} \cdot \frac{T(\mathbf{y}_i \to \mathbf{x}_i)}{T(\mathbf{x}_i \to \mathbf{y}_i)},\end{aligned} \tag{2.7}$$

where T denotes the transition probability between populations. In short, in the mutation move an MCMC move is performed at a specific chain independently, while other chains are kept unchanged. The irreducibility of Pop-MCMC is guaranteed by this mutation move.

2. Exchange move

The exchange move is the same as that used in PT. In this move, two different chains are randomly chosen first. And then the chromosomes of those two chains are exchanged to propose a new population. Let us suppose that the current population is $\mathbf{x}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_N\}$, and the ith and jth chains are selected Then, the newly proposed population will be $\mathbf{y}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_j, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N\}$. Similar to the mutation move, the new population is accepted according to the acceptance probability:

$$\alpha = \min(1, \gamma_e), \tag{2.8}$$

$$\begin{aligned}\gamma_e &= \frac{p^*(\mathbf{y}_{1:N})}{p^*(\mathbf{x}_{1:N})} \cdot \frac{T(\mathbf{y}_{1:N} \to \mathbf{x}_{1:N})}{T(\mathbf{x}_{1:N} \to \mathbf{y}_{1:N})} \\ &= \frac{p_i(\mathbf{x}_j)p_j(\mathbf{x}_i)}{p_i(\mathbf{x}_i)p_j(\mathbf{x}_j)}.\end{aligned} \tag{2.9}$$

The last equality holds due to the definition of the target distribution and the symmetry property of the transition probability.

In general, to obtain the higher acceptance rate, exchange moves are performed on chains that have similar target distributions with neighboring temperatures.

3. Crossover move

The crossover move is newly introduced in Pop-MCMC. The main concept of this move is borrowed from the genetic algorithm. The design of this move is the main contribution of Pop-MCMC. There are several variations of the crossover moves. One of the popular moves is the 1-point crossover move. The basic idea of it is as follows: As in the exchange move, two different chains, say ith and jth chains, are randomly selected. If the chromosome is a d-D vector, we randomly choose a natural number k between 1 and $(d-1)$. And new chromosomes $\mathbf{y}_i$ and $\mathbf{y}_j$ are proposed by swapping the same part of chromosomes $\mathbf{x}_i$ and $\mathbf{x}_j$ as follows.

$$\begin{bmatrix} \mathbf{x}_i = (x_{i1}, \cdots, x_{ik}, x_{i(k+1)}, \cdots, x_{id}) \\ \mathbf{x}_j = (x_{j1}, \cdots, x_{jk}, x_{j(k+1)}, \cdots, x_{jd}) \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{y}_i = (x_{i1}, \cdots, x_{ik}, x_{j(k+1)}, \cdots, x_{jd}) \\ \mathbf{y}_j = (x_{j1}, \cdots, x_{jk}, x_{i(k+1)}, \cdots, x_{id}) \end{bmatrix} \tag{2.10}$$

In this case, the ratio of proposal distributions in the acceptance probability is canceled by symmetry. We only need to calculate the ratio of the target distributions. A new population is proposed as $\mathbf{y}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{y}_i, \cdots, \mathbf{y}_j, \cdots, \mathbf{x}_N\}$, and according to the Metropolis-Hastings rule, it is accepted with probability

$$\alpha = \min(1, \gamma_c), \tag{2.11}$$

where

$$
\begin{aligned}
\gamma_c &= \frac{p^*(\mathbf{y}_{1:N})}{p^*(\mathbf{x}_{1:N})} \cdot \frac{T(\mathbf{y}_{1:N} \rightarrow \mathbf{x}_{1:N})}{T(\mathbf{x}_{1:N} \rightarrow \mathbf{y}_{1:N})} \\
&= \frac{p_i(\mathbf{y}_i) p_j(\mathbf{y}_j)}{p_i(\mathbf{x}_i) p_j(\mathbf{x}_j)} \cdot \frac{q(\mathbf{x}_i, \mathbf{x}_j | \mathbf{y}_i, \mathbf{y}_j)}{q(\mathbf{y}_i, \mathbf{y}_j | \mathbf{x}_i, \mathbf{x}_j)},
\end{aligned}
\tag{2.12}
$$

where $T(\mathbf{x}_{1:N} \rightarrow \mathbf{y}_{1:N})$ is $p(i, j|\mathbf{x}_{1:N}) \cdot q(\mathbf{y}_i, \mathbf{y}_j|\mathbf{x}_i, \mathbf{x}_j)$. $p(i, j|\mathbf{x}_{1:N})$ denotes the probability that ith and jth chains are chosen and $q(\mathbf{y}_i, \mathbf{y}_j|\mathbf{x}_i, \mathbf{x}_j)$ indicates the probability that the chromosomes $\mathbf{y}_i$ and $\mathbf{y}_j$ are proposed, when current chromosomes $\mathbf{x}_i$ and $\mathbf{x}_j$ are given. Choosing chains is independent of the current state, so $p(i, j|\mathbf{x}_{1:N})$ and $p(i, j|\mathbf{y}_{1:N})$ are canceled out in the second equality.

In order to include various ways of exchanging information between chromosomes, the 2-point crossover move, k-point crossover move, and adaptive crossover move were also proposed in the literature [24, 37].

2.3 Proposed Algorithm

In this chapter, we apply the Pop-MCMC method to MRF optimization. For this purpose, we design new effective 2-D mutation and crossover moves to explore the high dimensional state space efficiently.

Given a target probability distribution $p(\mathbf{x}) \propto \exp\{-E(\mathbf{x})\}$, our aim is to find the state $\mathbf{x}$ where the probability is maximized. In Pop-MCMC, we draw multiple samples from multiple chains at the same time with respect to the following distributions.

$$
p_i(\mathbf{x}_i) = p(\mathbf{x}_i)^{\frac{1}{T_i}} \propto \exp\left\{-\frac{E(\mathbf{x}_i)}{T_i}\right\},
\tag{2.13}
$$

where T_i is the temperature of ith chain. The appropriate sequence of the tempera-

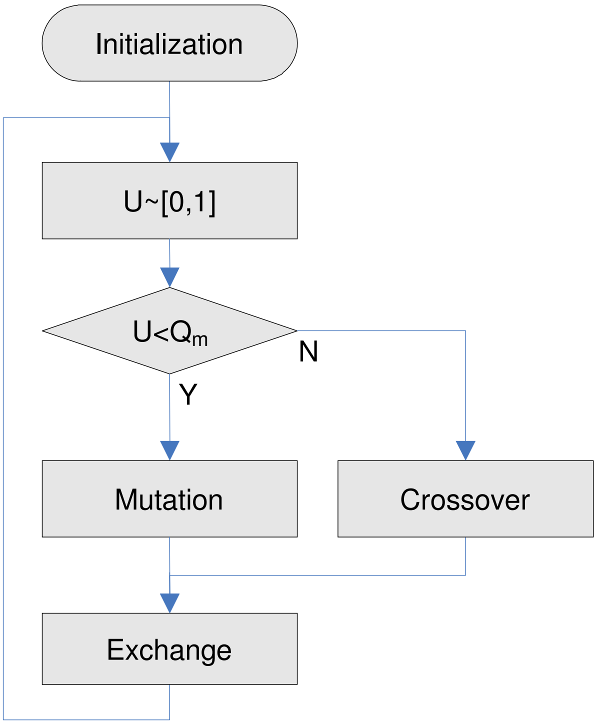


Figure 2.2: The overall flow chart of the proposed Pop-MCMC algorithm applied to stereo matching.

tures can be designed empirically according to the target distribution. Each sample from each chain is a chromosome, and chromosomes interact with each other, which helps perform global moves.

The overall flow of Pop-MCMC is illustrated in Figure 2.2. The three moves, mutation, crossover, and exchange moves are repeatedly performed and samples are generated at each iteration. In this process, we first choose a random number U between 0 and 1, and compare U with the mutation rate Q_m. Depending on the value of U, we choose one move between mutation and crossover. So, by varying Q_m, we can control the rates between the global move (crossover) and local move (mutation) easily. This means that Q_m adjusts the trade-off between exploration and convergence of the algorithm [38]. A proper value of Q_m can be chosen according to the given problem, the model, or the number of chains. For example, if a large number of chains are used, Q_m is usually set to a small value for faster convergence. Let us describe the detailed design of each move for the Gibbs distribution of MRF.

1. Mutation move

In the proposed algorithm, we employ the MCMC kernel of SWC-2 for the mutation move of a randomly selected chain. At first, we construct a random cluster $\mathcal{V}_0$ as in SWC-2 for a selected chain. For clustering, we need to design edge probability q_e, which determines whether the edge should be retained or not. We define the edge probability,

$$q_e = 1 - \exp\left(- \frac{K_i \cdot \mathcal{S}(s,t)}{\dfrac{\theta_{v_1}(x_{v_1})}{|\mathcal{N}_{v_1}|} + \dfrac{\theta_{v_2}(x_{v_2})}{|\mathcal{N}_{v_2}|} + 2} \right), \qquad (2.14)$$

where v_1 and v_2 represent neighboring nodes, $|\mathcal{N}_v|$ is the number of the pixels in the node (segment) v, and K_i represents a weighting factor for the chosen ith chain. The more similar the intensities of the connected nodes and the lower the matching costs are, the higher the probability that the edge remains. Note that the matching costs are normalized by the sizes of the corresponding segments. By varying K_i, we can control the average size of clusters. A bigger K_i tends to generate bigger clusters. We set K_i to increase as i increases. Consequently, clusters are likely to be small in lower-temperature chains and big in higher-temperature chains. It helps more effective exploration and also prevents chromosomes from correlating with each

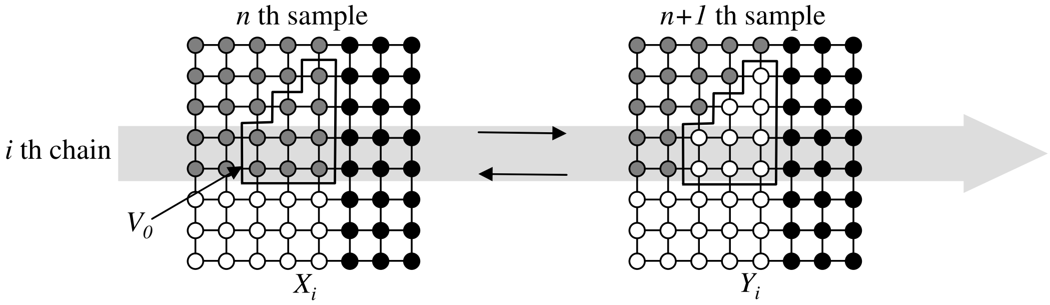


Figure 2.3: An example of mutation move.

other.

The new label l' for the selected cluster $\mathcal{V}_0$ is proposed according to the following proposal distribution.

$$q(l' \mid \mathcal{V}_0, \mathbf{x}_i) = \exp\Bigg[-\Bigg\{\frac{\sum_{v\in\mathcal{V}_0}\theta_v(l')}{\sum_{v\in\mathcal{V}_0}|\mathcal{N}_v|} + 1 - \prod_{\langle v_1,v_2\rangle\in N, v_1\in\mathcal{V}_0, v_2\notin\mathcal{V}_0} \mathbf{1}(l' = f_{v_2})\Bigg\}\Bigg], \tag{2.15}$$

where l' is the newly proposed label for $\mathcal{V}_0$, and $\mathbf{x}_i$ is the current state of selected ith chain. When the nodes in the cluster $\mathcal{V}_0$ have low matching costs and there exist neighboring nodes of same label, the value of $q(l' \mid \mathcal{V}_0, \mathbf{x}_i)$ becomes high. After a new label is proposed, it is accepted according to the Metropolis-Hastings rule. By substituting (2.13) and the transition probability in (2.3) into (2.7), we can calculate the acceptance probability:

$$\begin{aligned}\alpha &= \min(1, \gamma_m)\\ &= \min\left(1, \frac{p_i(\mathbf{y}_i)}{p_i(\mathbf{x}_i)} \cdot \frac{T(\mathbf{y}_i \to \mathbf{x}_i)}{T(\mathbf{x}_i \to \mathbf{y}_i)}\right)\\ &= \min\left(1, \exp\left\{\frac{E(\mathbf{x}_i) - E(\mathbf{y}_i)}{T_i}\right\} \cdot \frac{q(\mathcal{V}_0 \mid \mathbf{y}_i)q(l \mid \mathcal{V}_0, \mathbf{y}_i)}{q(\mathcal{V}_0 \mid \mathbf{x}_i)q(l' \mid \mathcal{V}_0, \mathbf{x}_i)}\right),\end{aligned} \tag{2.16}$$

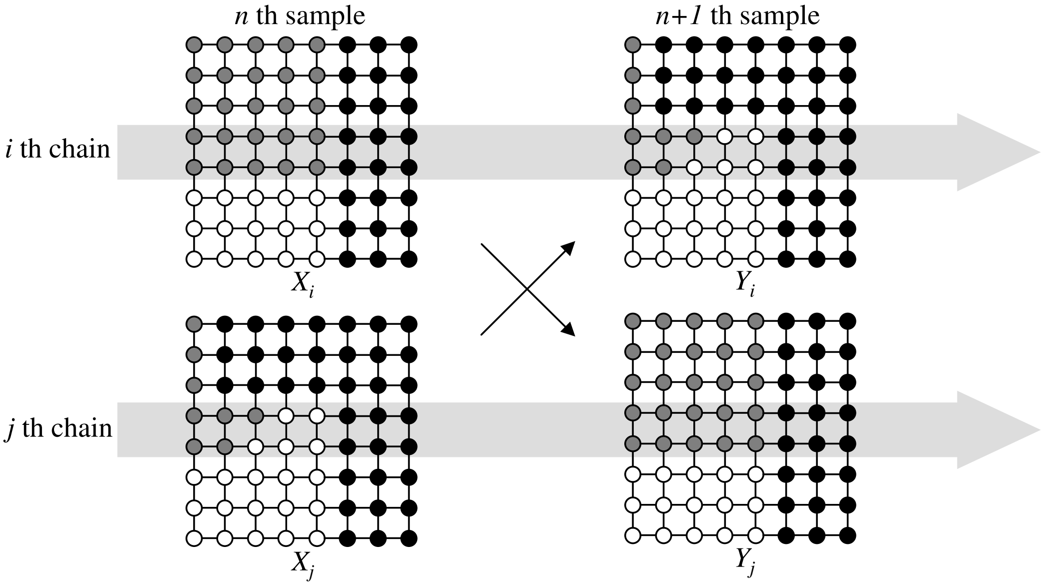


Figure 2.4: An example of exchange move.

where $\mathbf{y}_i$ is the proposed state of the ith chain, and $q(\mathcal{V}_0 \mid \mathbf{x}_i)$ is the probability for selecting cluster $\mathcal{V}_0$ when current state is $\mathbf{x}_i$. Figure 2.3 illustrates an example of mutation move on the ith chain.

2. Exchange move

In this move, we choose two chains and propose to exchange the chromosomes of two chains. The proposal is accepted or not by the Metropolis-Hastings rule. Figure 2.4 shows an example of exchange move. Note that for the exchange move, there is no need for a special design for stereo matching problem. So, when the ith and jth chains are selected, by substituting (2.13) into (2.9), we can obtain the acceptance probability by

$$\begin{aligned}\alpha &= \min(1, \gamma_e) \\ &= \min\left(1, \frac{p_i(\mathbf{x}_j)p_j(\mathbf{x}_i)}{p_i(\mathbf{x}_i)p_j(\mathbf{x}_j)}\right) \\ &= \min\left(1, \exp\left[\left\{E(\mathbf{x}_i) - E(\mathbf{x}_j)\right\}\left(\frac{1}{T_i} - \frac{1}{T_j}\right)\right]\right),\end{aligned} \tag{2.17}$$

where $\mathbf{x}_i$ and T_i are the current state and temperature of the ith chain. In order to achieve faster mixing rate, we need to raise the acceptance rate, and this can be accomplished by choosing two neighboring chains that have similar temperatures. Then, from the above equation, the Metropolis-Hastings ratio tends to get bigger.

3. Crossover move

Typical crossover moves commonly used in conventional Pop-MCMC are the 1-point crossover and 2-point crossover moves. However, since these methods are designed for the chromosomes of 1-D vectors, it is inappropriate to apply them directly to the stereo matching problem, in which the chromosomes are 2-D state configurations. Nonetheless, the 1-point and 2-point crossover moves have an advantage of low computational complexity because the most of terms in the Metropolis-Hastings ratio cancel out each other. Therefore, in this work, we introduce a new 2-D crossover move that maintains this advantage. Detailed description of the proposed crossover move is as follows.

We first choose two chains randomly and construct a cluster $\mathcal{V}_0$ in a similar way as in SWC-2 (or the mutation move). However, there are two differences in constructing $\mathcal{V}_0$ compared with SWC-2. First, q_e is set constant, not adaptively determined with

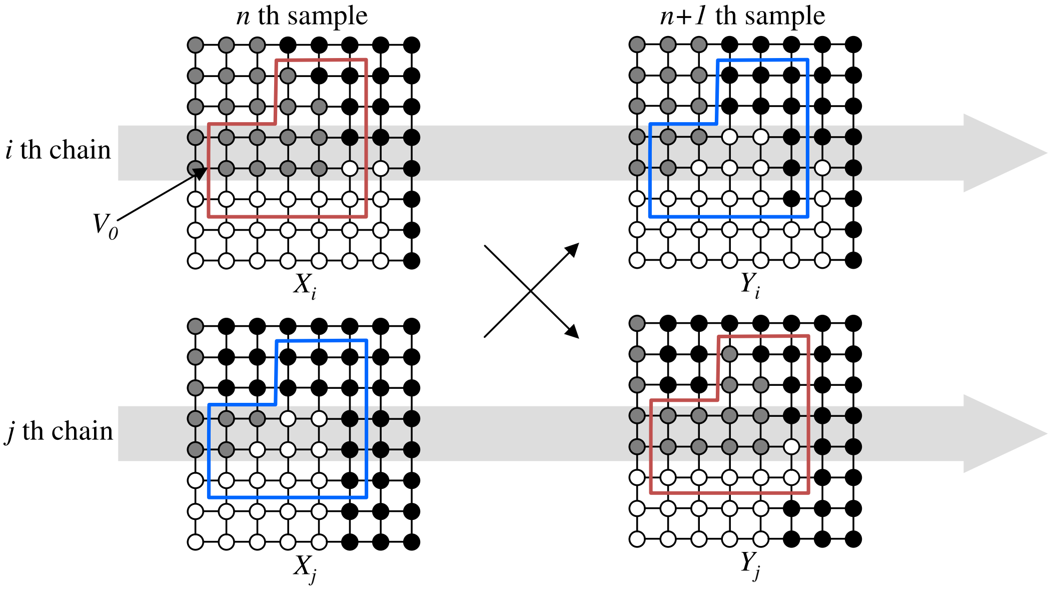


Figure 2.5: An example of crossover move.

the matching costs or the intensities of the input image, since there is no need for the nodes of the cluster $\mathcal{V}_0$ to be homogeneous in this case. It is also computationally efficient to use q_e as a constant value because the proposal distribution part in the Metropolis-Hastings ratio is canceled out. Second, when we calculate the probability q_e, we do not have to check whether the labels of the nodes are the same or not, so the resulting cluster $\mathcal{V}_0$ can have nodes with different labels. Therefore, compared with the mutation move that requires the identifying and removing processes of all the edges connecting the nodes with different labels, the selecting scheme and the calculation of the acceptance probability of $\mathcal{V}_0$ in the crossover move is much simpler. Eventually this property enables high efficiency in computation, and also the freedom in the construction of $\mathcal{V}_0$ helps to achieve faster convergence.

The process after constructing a cluster $\mathcal{V}_0$ is similar to the 1-point crossover move. From the chromosomes $\mathbf{x}_i$ and $\mathbf{x}_j$ of two selected chains, new chromosomes

$\mathbf{y}_i$ and $\mathbf{y}_j$ are proposed by exchanging the labels of the nodes which belong to the cluster $\mathcal{V}_0$ as shown in Figure 2.5. The acceptance probability $\alpha = \min(1, \gamma_c)$ of the newly proposed chromosomes is calculated, and the next population of samples is determined. By substituting equation (2.13) into the Metropolis-Hastings rule in (2.12), we can obtain γ_c as follows.

$$\begin{aligned}\alpha &= \min(1, \gamma_c)\\ &= \min\left(1, \frac{p_i(\mathbf{y}_i)p_j(\mathbf{y}_j)}{p_i(\mathbf{x}_i)p_j(\mathbf{x}_j)} \cdot \frac{q(\mathbf{x}_i, \mathbf{x}_j|\mathbf{y}_i, \mathbf{y}_j)}{q(\mathbf{y}_i, \mathbf{y}_j|\mathbf{x}_i, \mathbf{x}_j)}\right)\\ &= \min\left(1, \frac{p_i(\mathbf{y}_i)p_j(\mathbf{y}_j)}{p_i(\mathbf{x}_i)p_j(\mathbf{x}_j)}\right)\\ &= \min\left(1, \exp\left[\frac{E(\mathbf{x}_i) - E(\mathbf{y}_i)}{T_i} + \frac{E(\mathbf{x}_j) - E(\mathbf{y}_j)}{T_j}\right]\right),\end{aligned} \tag{2.18}$$

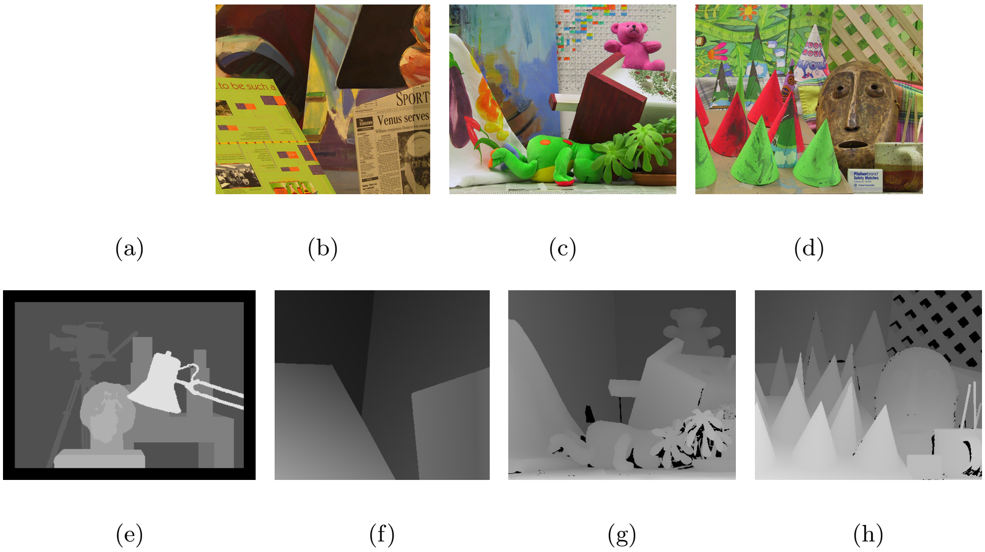
where we used the symmetric property of the proposal distribution $q(\mathbf{x}_i, \mathbf{x}_j|\mathbf{y}_i, \mathbf{y}_j)$.

The proposed Pop-MCMC algorithm is summarized in Algorithm 1.

2.4 Experiments

2.4.1 Segment-based stereo matching

In order to improve the accuracy of the disparity map, various energy models have been newly proposed for the stereo problem. Among them, we choose the segment-based energy model since it is known as one of the best energy models and it is robust to noise [39, 40, 41, 42]. This model assumes that each segment corresponds to a planar patch in the scene. In a segment-based energy model, the reference image is first over-segmented. This segment-based energy model also reduces running time since the number of nodes is much smaller than pixel-based model. Mean-shift


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.6: Test stereo images: (a)-(d) reference images, (e)-(h) ground truth disparity maps. (a, e) Tsukuba, (b, f) Venus, (c, g) Teddy, and (d, h) Cones.

algorithm is often used for the segmentation [43].

Each segment is defined as a node $v \in \mathcal{V}$, and neighboring nodes s and t are connected with edges $\langle s, t\rangle \in \mathcal{E}$. Then we construct a graph $\mathbf{G} = (\mathcal{V}, \mathcal{E})$. And the energy function is defined by

$$\begin{aligned} E(\mathbf{x}) &= \sum_{s\in\mathcal{V}} \theta_s(x_s) + \sum_{\langle s,t\rangle\in\mathcal{E}} \theta_{st}(x_s, x_t) \\ &= \sum_{v\in\mathcal{V}} C_{\text{SEG}}(f_v) + \sum_{\langle s,t\rangle\in N} \beta_{s,t}\mathbf{1}(f_s \neq f_t), \end{aligned} \tag{2.19}$$

where $\mathbf{x}$ represents the current state of every segment, f_v is an estimated plane for each segment, $C_{\text{SEG}}(f_v)$ is a matching cost, and $\beta_{s,t}$ is a penalty for different neighboring nodes of s and t, which are defined by

$$C_{\text{SEG}}(f_v) = \sum_{(x,y)\in\mathcal{V}} C(x, y, f_v(x, y)), \tag{2.20}$$

$$\beta_{s,t} = \gamma \cdot BL(s, t) \cdot \mathcal{S}(s, t), \tag{2.21}$$

where function $C(x, y, f_v(x, y))$ is the Birchfield-Tomasi cost, $BL(s, t)$ is the shared border length, and $\mathcal{S}(s, t)$ is the mean color similarity defined by

$$\mathcal{S}(s, t) = \frac{1}{2}\left(1 - \min\left(1, \frac{|R_{\mathcal{V}_s} - R_{\mathcal{V}_t}| + |G_{\mathcal{V}_s} - G_{\mathcal{V}_t}| + |B_{\mathcal{V}_s} - B_{\mathcal{V}_t}|}{255}\right)\right) + \frac{1}{2}, \tag{2.22}$$

where $R_{\mathcal{V}_s}$, $G_{\mathcal{V}_s}$ and $B_{\mathcal{V}_s}$ are average intensity values of segment $\mathcal{V}_s$, which are between 0 and 255. Mean color similarity has a value between $\frac{1}{2}$ and 1. When two neighboring segments have similar intensities, it becomes closer to 1. By varying γ, we can control the relative effect of matching cost and smoothness cost.

We first need to make a list of the planes for assigning each segment to a plane by examining segment by segment. For each pixel, we calculate the initial disparity by using SAD (Sum of Absolute Differences) and WTA (Winner Takes All) schemes. Using these initial disparities, we fit a plane for each segment. The equation of a plane in 3D-space can be written by

$$d(x, y) = c_1x + c_2y + c_3, \tag{2.23}$$

where x and y are the coordinates of a pixel, and $d(x, y)$ is its disparity. Based on the above equation, we construct the following algebraic equation for each segment.

$$\mathbf{A}\,[c_1, c_2, c_3]^{\mathrm{T}} = \mathbf{B}, \tag{2.24}$$

where the ith row of the matrix $\mathbf{A}$ is the coordinates $[x_i, y_i, 1]$ of the ith pixel, and the ith row of the matrix $\mathbf{B}$ is the disparity $d(x_i, y_i)$ of that pixel. Then, the values of c_1, c_2, c_3 are obtained as a least squares solution by solving (2.24). In this method, the outlier disparities are initially detected and removed by a disparity crosschecking method [41]. Once we find the plane parameters, we can further identify more outlier disparities that are not close to the fitted plane. For those pixels with outlier disparities, we re-estimate the correct disparities by confining the search range to be small near the fitted plane. Then, the least squares method is repeated to update parameters c_1, c_2, c_3 based on the modified disparities.

The above plane fitting process is repeated for each segment and newly found planes are added to a list. After that, each segment is assigned to a plane in the list that has lowest C_{SEG} value. Then we group the segments assigned to the same plane. And for each group, the above plane fitting is repeated in order to improve the accuracy of a plane. At last, we have the final list of the planes to use. Although, this plane-based model does not explicitly handle the occlusion, occluded pixels are likely to be detected as outlier through the crosscheck in plane estimation.

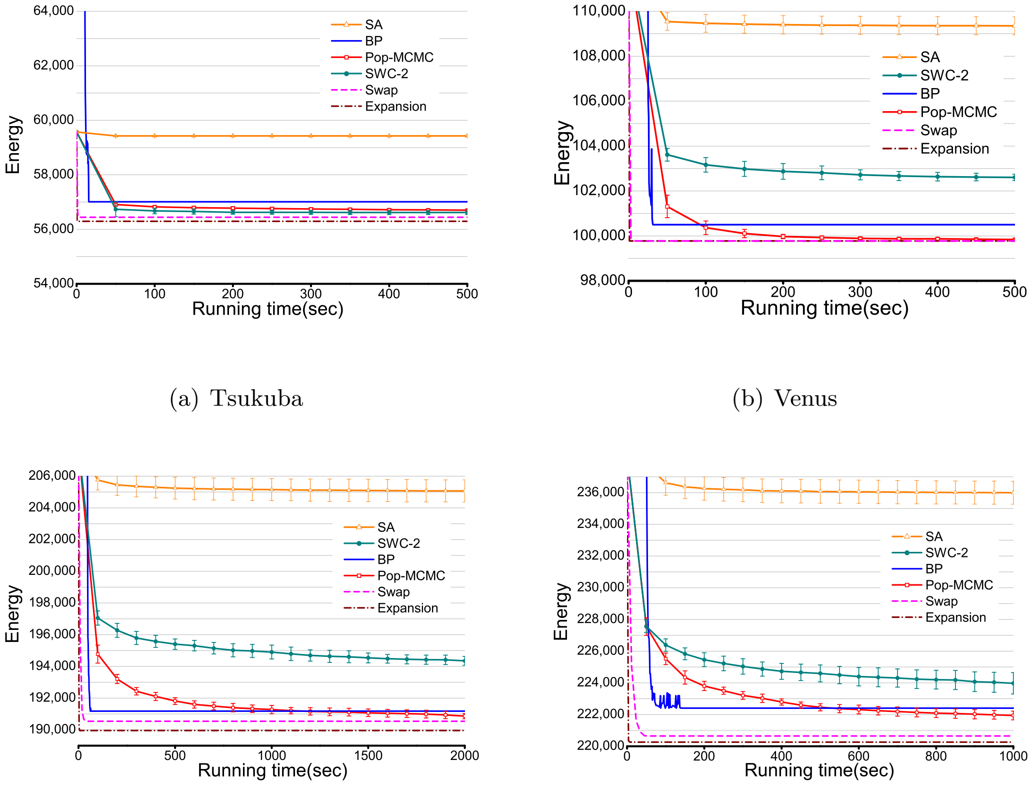
We have implemented the proposed algorithm on a 2.8GHz Pentium IV PC platform. In this section, we evaluate the performance of the proposed algorithm by comparing with other conventional methods such as SWC-2, SA, BP, and Graph cuts. In addition, we illustrate the effects of each move, temperature parameter, and the number of chains. We tested the proposed algorithm on several benchmark images in the Middlebury datasets [1]. Figure 2.6 shows the reference images and the ground truth maps of the test images. We used the segment-based energy model in (2.19) for the test. Pop-MCMC, SWC-2, and SA methods were repeated ten times on each test stereo image pair since they are stochastic methods, and the averages and standard deviations of the resulting energies were compared.

We fixed the parameter values of Pop-MCMC for all the test sets. Empirically, the temperatures were set to be decreasing linearly in the range of predefined maximum and minimum temperature values. The maximum and minimum temperatures were set to 1.0 and 0.0001, respectively, and the number of chains was set to five. Q_m was set to be 0.25. For the edge probability of the ith chain, we set $K_i = 3i + 1$. This helped the chromosomes not to correlate with each other.

Figure 2.7 presents the comparison of the energy plots against running time in second for Pop-MCMC, SWC-2, SA, BP, and Graph cut methods (expansion move and swap move). The same energy model was applied to each method. For the implementation of SWC-2, we followed the work of Barbu and Zhu [21, 22], and for Graph cuts, we used the source code from [44]. Expansion move method showed the best performance in all the tests. The proposed Pop-MCMC algorithm was comparable to the expansion and swap move methods. Although proposed method perform slightly worse than expansion and swap move methods, Pop-MCMC has much wider applicability than expansion and swap move methods. Contrary to expansion and swap methods which can be applied only to submodular functions with pairwise priors, Pop-MCMC can be applied to any type of energy functions even including higher-order MRFs and highly complicated MRFs. Therefore Pop-MCMC can be a good alternative to expansion and swap move methods. And, Pop-MCMC algorithm reached much lower energy states than SA and SWC-2 on all the cases except Tsukuba. Note that it even showed better performance than BP for all the test images. On the Tsukuba images, each method obtained relatively good result since the dimension of the solution space is low and thus the energy model is relatively simple. Note that on all test images, the convergence rates of Pop-MCMC are much faster and its standard deviations are consistently smaller than those of SA and SWC-2. From these results we can argue that conventional sampling-based methods like SA and SWC-2 are easily trapped at local minima, while Pop-MCMC is more likely to approach the global minimum due to the global moves in Pop-MCMC, such as exchange and crossover moves.

Algorithm 1 Proposed Pop-MCMC algorithm

```
(Initialize)
Initialize the population $\mathbf{x}_{1:N}$ by Winner-Takes-All manner with data cost.
Set the temperatures $T_1 < T_2 < \cdots < T_N$.
repeat
  if $U \sim [0,1] < Q_m$ then
    for $i = 1$ to $N$ do
      (Mutation)
      Select a random node $v$ in $i$th chain.
      Draw a cluster from a node $v$ with SWC-2.
      Propose a new label for the cluster and determine whether accept it or not
      with Metropolis-Hastings rule.
    end for
  else
    for $i = 1$ to $\lfloor \frac{N}{5} \rfloor$ do
      (Crossover)
      Select two random chains and a random node $v$.
      Draw a cluster from node $v$ with modified SWC-2.
      Determine whether swap the cluster or not with Metropolis-Hastings rule.
    end for
  end if
  for $i = N - 1$ to 1 do
    (Exchange)
    Perform the exchange move onto $i$th and $i + 1$th chains with Metropolis-
    Hastings rule.
  end for
until The algorithm converges.
```



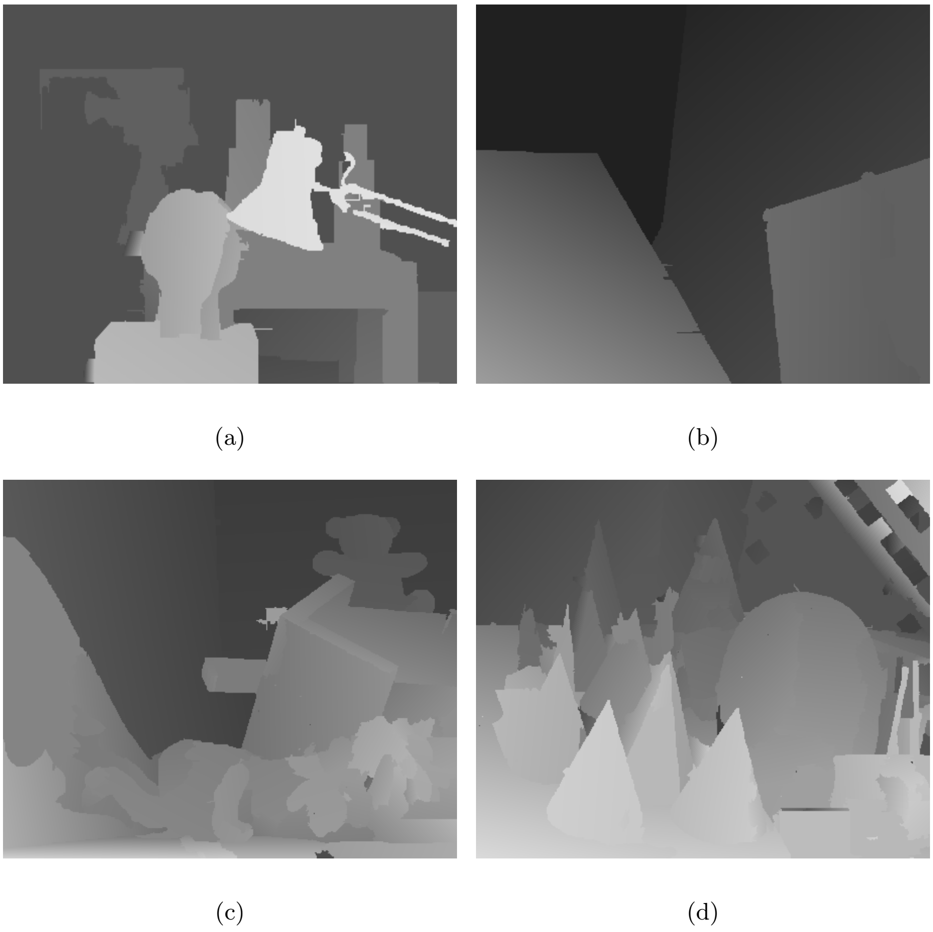
(a) Tsukuba

(b) Venus

(c) Teddy

(d) Cones

Figure 2.7: Performance (energy vs. running time) comparison of Pop-MCMC, SA, SWC-2 and BP on (a) Tsukuba, (b) Venus, (c) Teddy, and (d) Cones. Pop-MCMC obtains lower energy results than other methods except on Tsukuba.



(a)

(b)

(c)

(d)

Figure 2.8: Results of the proposed algorithm: the disparity maps of (a) Tsukuba, (b) Venus, (c) Teddy, and (d) Cones.

Table 2.1: The error rates for each test image [1]. For the sampling-based methods, we denote the average and standard deviation for ten trials. *nonocc*, *all*, and *disc* represent the error rate within non-occluded region, the whole image, and the vicinity of discontinuity, respectively.

method	Tsukuba			Venus			Teddy			Cones		
	nonocc	*all*	*disc*	*nonocc*	*all*	*disc*	*nonocc*	*all*	*disc*	*nonocc*	*all*	*disc*
Pop-MCMC	3.35 (±0.42)	3.88 (±0.42)	10.3 (±0.92)	0.22 (±0.01)	0.35 (±0.02)	2.89 (±0.17)	12.0 (±0.56)	17.9 (±0.69)	21.7 (±0.63)	13.3 (±0.37)	19.2 (±0.54)	23.7 (±0.57)
SWC	3.69 (±1.24)	4.28 (±1.23)	10.4 (±1.21)	0.9 (±0.27)	1.1 (±0.26)	5.57 (±0.11)	11.6 (±0.72)	17.8 (±0.86)	22.2 (±0.99)	13.5 (±0.72)	20.3 (±0.97)	23.4 (±0.76)
SA	3.5 (±0.28)	4.09 (±0.3)	9.58 (±0.48)	0.94 (±0.16)	1.34 (±0.21)	7.67 (±0.78)	14.8 (±0.61)	21.4 (±0.59)	24.3 (±0.79)	15.6 (±0.69)	22.9 (±0.78)	25.1 (±0.43)
BP	3.12	3.76	10.5	0.21	0.34	2.81	10.5	16.5	20.4	12.9	19.2	23.3
Expansion	4.12	4.73	12.2	0.21	0.34	2.81	10.9	12.4	19.1	12.5	18.6	23.1
Swap	2.56	3.09	9.15	0.21	0.34	2.81	10.5	12.0	19.7	13.0	19.0	23.6

The disparity error rates of the Pop-MCMC were compared with those of other algorithms and shown in Table 2.1, and the resulting disparity maps of the proposed algorithm are shown in Figure 2.8. Note that there are some limitations of the segment-based energy model. When real world objects are piecewise planar, the results are quite good. However, for the cases of Teddy and Cones that include objects with curved surfaces, the performance seems not satisfactory. And also, for a fronto-parallel plane, a non-segment based energy model can be superior to the segment-based energy model due to the smaller number of labels. In addition, since occlusion or visibility was not considered in our stereo model, the error rates at the vicinity of discontinuity were relatively large.

2.4.2 Parameter analysis

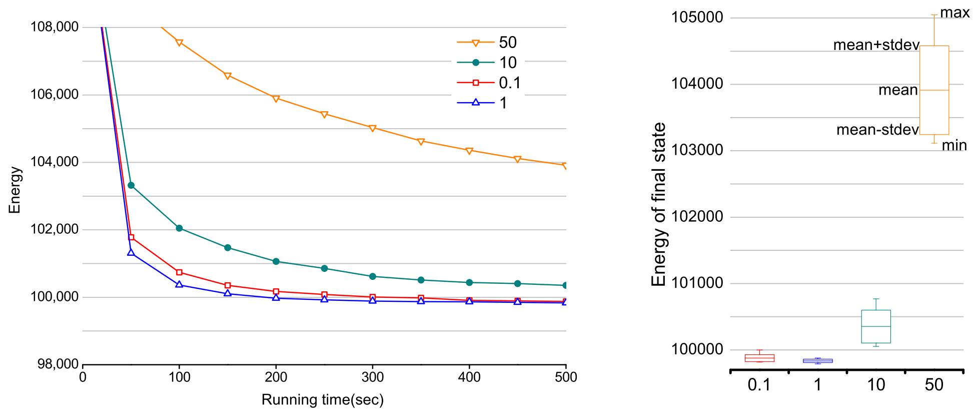
Figure 2.9 and Figure 2.10 exhibit the performance for differing parameters. Both experiments were carried on the Venus image. Figure 2.9 shows the energy convergence plots according to the variation of the max-temperature. The min-temperature was set to 0.0001. We observed that if the max-temperature was too low, it quickly moved to the nearest minimum but easily got stuck in local minima. While if it was too high, the algorithm was rarely trapped in local minima but the convergence speed became too slow. Figure 2.10 shows the energy convergence plots by varying the number of chains. If the population size was large, it helped each other to reach the global minimum by exchanging information. However, a large size of population usually increases redundancy in the algorithm. We found empirically that for our segment-based stereo energy model, the optimal max-temperature was 1.0 and the optimal number of chains was five.

Figure 2.11 and Figure 2.12 report the contribution of each move in Pop-MCMC.

Figure 2.11 shows the statistics of each move while Pop-MCMC is running on the Venus images. We counted the number of accepted moves every ten seconds. As shown in the graphs, the mutation move occurred most frequently. In the beginning, all the three moves frequently occurred but as time went on, they tended to decrease since they were approaching the optimum. The exchange move occasionally occurred when higher-temperature chromosomes had lower energy states than those of lower-temperature chromosomes. Figure 2.12 compares the energy convergence rates for different combinations of moves. We illustrated the energy curves and the boxplots of the final state energies. We performed the experiment on the Venus images. The exchange move contributed larger amount than the crossover move. Obviously, when we combined all the three moves, they together helped each other to achieve fast convergence. Boxplots of the final state energies show not only that the three moves together could reach lower energy state, but also that they decreased the standard deviation, and in turn made the algorithm quite stable. It took about 190 seconds to minimize the energy to be 100,000 using all moves. However, if one of the moves was missing, it became much slower. For example, without the crossover move, it took about 440 seconds, and without the exchange move, it could not reach that state until 500 seconds.

2.5 Summary

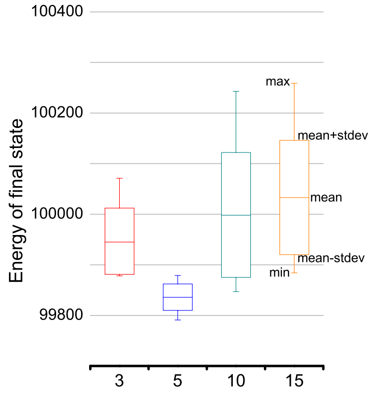
In this chapter, we proposed a new stereo matching algorithm based on Pop-MCMC. We showed that the proposed sampling-based Pop-MCMC was a good optimizer for stereo problem. Pop-MCMC uses multiple chains in parallel, and establishes faster mixing rate by exchanging information between chromosomes. In this work,



(a) (b)

Figure 2.9: The performance of the Pop-MCMC for different max temperature values: (a) Energy curves, (b) boxplots of the final states.

we designed new effective 2-D mutation and crossover moves for stereo matching based on cluster sampling technique. Consequently, it is shown that the proposed algorithm provides much faster convergence rate than conventional sampling-based methods including SA and SWC, and gives lower energy states than BP. We also investigated the contribution of each move. Combining all the three moves together made the algorithm more stable. In addition, we analyzed the effect of parameters such as temperature and the number of chains, and found the optimal parameters for our problem. We have a plan to apply and analyze the performance of the proposed method to more sophisticated stereo energy models including occlusion handling and visibility terms as well as the segmentation problem.



(a) (b)

Figure 2.10: The performance of the Pop-MCMC for different number of chains: (a) Energy curves, (b) boxplots of the final states.

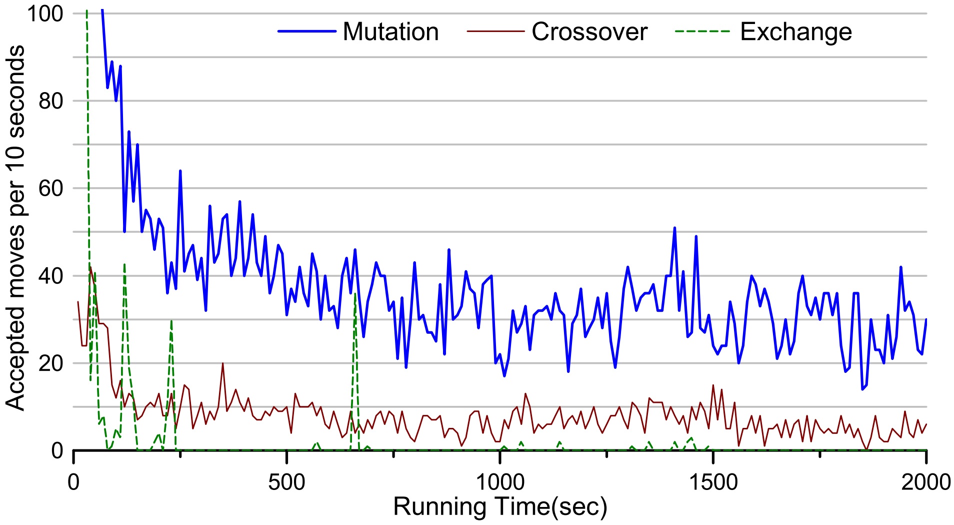
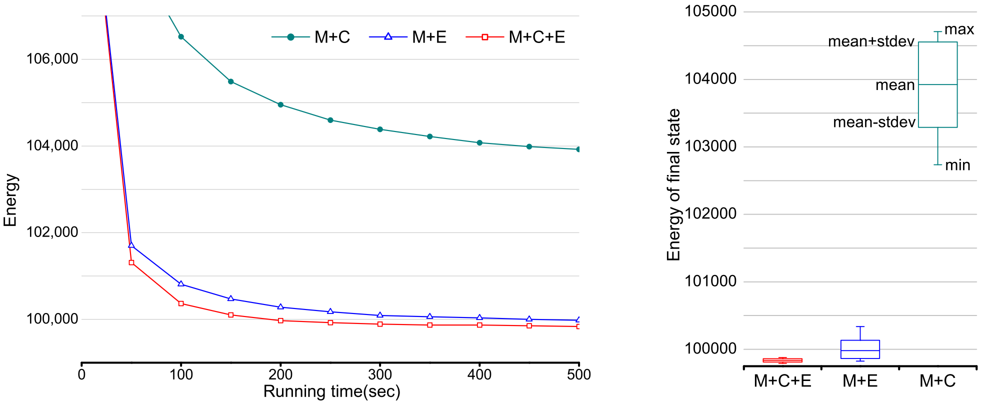


Figure 2.11: The statistics of each move while running the algorithm on the Venus images. Initially, all the three moves are quite active, and then tend to decrease as time goes on. While mutation and crossover moves consistently occur, exchange move occurs occasionally.



(a) (b)

Figure 2.12: The performance of the Pop-MCMC for different combinations of moves: (a) Energy curves, (b) boxplots of the final states.

Chapter 3

Markov Chain Monte Carlo Combined with General Deterministic Methods

3.1 Introduction

Markov Random Field (MRF) models are of fundamental importance in computer vision. Many vision problems have been successfully formulated in MRF optimization. They include stereo matching, segmentation, denoising, and inpainting, to mention just a few. Recently, Szeliski *et al.* [2] presented a comprehensive review of the standard MRF-based vision problems and the comparative results of existing optimization algorithms.

Many algorithms for minimizing the aforementioned energy function have been proposed. Although those methods have been successively applied to various problems, the story becomes different when it comes to more difficult MRF problems.

There are some known factors which make MRF problems more difficult: non-submodular functions, strongly coupled MRF models, high connectivity and higher-order clique potentials. It is known that more non-submodular terms make the problem harder [15]. The coupling strength also affects performance in solving MRF problems. The coupling strength refers to the relative strength of pairwise versus unary terms. As coupling strength increases, problems become more difficult [2, 19]. High connectivity of graphs is another factor which makes the problem difficult [20]. Higher-order clique potential also make the problem difficult. Despite difficulty, higher-order clique potential has often been used to improve the results in some vision applications [6, 5]. More difficult MRF models are inevitable to incorporate realistic image priors into the models. (*e.g.*, occlusion terms in stereo and texture information in denoising and segmentation.) To those difficult examples, most existing algorithms are not applicable, and even with some applicable algorithms the results are far from the global optimum.

Therefore, we definitely need a more efficient optimization technique to cope with such difficult MRF vision problems. Our main idea is to combine the stochastic sampling and deterministic algorithms so that we can take advantages of both sides. Our new algorithm is mainly inspired by the work of Strens *et al.* [45]. They used direct search optimization (downhill simplex method and differential evolutions) in the framework of Population-based Markov Chain Monte Carlo (Pop-MCMC) to increase stochastic sampling performance. Although they improved performance by combining sampling and optimization method, they remain focused only on sampling rather than optimization. Moreover, they did not provide a general framework for combination of algorithms. On the other hand, we propose a general framework for optimization which is suitable for many vision applications. This chapter is organized as follows. We continue with recent researches which are related to our work in the following section. We present the details of the proposed algorithm in Section 3.3. Section 3.4 gives the experimental results both on synthetic and real problems.

3.2 Related works

Many algorithms have been proposed to solve MRF problems. The existing algorithms can be divided into two approaches: deterministic and stochastic sampling algorithms. Some of the well-known deterministic algorithms are move-making algorithms. Move-making algorithms iteratively make local moves to explore the solution space. They include Iterated Conditional Modes (ICM), the Gradient Descent Algorithm and Graph Cuts [12, 13, 14]. Graph Cuts are the state of the art among those move-making algorithms. It becomes more powerful due to recent advances including the fusion move and the Quadratic Pseudo-Boolean Optimization (QPBO) algorithm [15, 5]. Graph Cuts iteratively optimize the binary sub-problems of the original problem. They are fast, accurate and even find global optima for some classes of functions. Another important class of deterministic algorithms is the message passing approach. It includes Belief Propagation (BP) [16] and Tree Reweighted Message Passing (TRW) [17, 18]. BP was originally developed for graphs without cycles. Although there is no guarantee of convergence in the case of the graph with cycles, it has been successfully applied to vision problems. One of the important properties of TRW is that it gives a lower bound of the energy function, which can be used to check optimality of the solutions.

Sampling-based algorithms have also been applied to the MAP–MRF based vision problems. They include Markov Chain Monte Carlo (MCMC) algorithm and its variants. MCMC is one of the most popular sampling algorithms. It was originally developed to generate samples from a given target distribution or to integrate functions in high dimensional spaces. Along with Simulated Annealing, MCMC has also been used to obtain optimum samples of target functions. In MRF optimization for vision, Swendsen–Wang Cuts was proposed to solve image segmentation and stereo problems [22, 46]. Recently, Kim *et al.* [27, 28] proposed a more advanced MCMC method called Pop-MCMC to optimize a plane-based stereo energy model. In addition, Jung *et al.* [47, 48] proposed window annealing algorithm to increase mixing ratio of the MCMC method.

Aforementioned methods have been successively applied to many problems. Nevertheless, they all are unsatisfactory when it comes to more difficult MRF problems. Note that α-Expansion and $\alpha\beta$-Swap is able to achieve satisfactory results only with submodular energy functions, whose pairwise terms satisfy

$$\theta_{st}(\beta,\gamma) + \theta_{st}(\alpha,\alpha) \leq \theta_{st}(\beta,\alpha) + \theta_{st}(\alpha,\gamma). \tag{3.1}$$

To solve functions whose pairwise terms does not satisfy this relationship, we should truncate the non-submodular terms, *i.e.* violating terms should be replaced with submodular approximations. Consequently, it seriously degrades the quality of solutions when violating terms are getting larger. On the other hand, QPBO can be applied those functions without truncation. As the difficulty of problems increases, however, it produces more unlabeled pixels, which yields unsatisfactory results. The number of unlabeled pixels depends on the strength of unary and pairwise terms, the number of non-submodular terms and the connectivity of the graph structure [15]. To resolve this problem, probing is proposed by Rother *et al.* [15] but it still leaves some pixels unlabeled. In addition, all Graph Cut based algorithm can handle only pairwise graphs. Message passing algorithms are also degraded as the difficulty of the problem increase. The complexity of belief propagation exponentially increases with the size of the largest clique. Also, Komodakis and Paragios [19] mentioned that the solutions and the lower bounds do not converge in difficult MRF problems. Note that the gap between the solutions and the lower bounds of TRW-S (Sequential TRW) [17] can be an efficient measure of qualities of the solutions. Sampling-based algorithms also have weaknesses. Although applicable to any class of MRF problems, they are usually slower than deterministic algorithms even in the simple MRF problems [27, 28], and do not lower the energy state substantially [47, 48]. If difficulty of the problem increases, we do not think they can solve the problems in a practical timescale since they are too slow even in the simple ones.

Recently, there has been increasing emphasis on the higher-order MRF models because it can capture the rich statistics of natural scenes [4, 5, 8, 9, 49]. However, due to intrinsic difficulty of the model and the lack of efficient algorithms, it has often been troublesome to use the higher-order MRF models. There are some approaches to overcome those limitations. First, in the move-making algorithms, the reduction technique has been introduced. This technique reduces higher-order clique potentials into pairwise ones so that it can be possible to apply the algorithms for only up to pairwise ones, such as Graph Cuts. Kolmogorov and Zabih introduced a technique that reduces third-order clique potentials into pairwise ones [14]. Unfortunately, their reduction was limited only up to the third-order cliques. Ali *et al.* recently used more general reduction technique, which can reduce any order clique potentials into pairwise ones [10]. This technique, however, produces severe amount of non-submodular term. Because of that, it is hard to apply this technique to the clique potentials which has higher-order than 3. More recently, Ishikawa proposed a new reduction technique which reduces any order clique potentials into pairwise ones [9, 8]. However, we cannot still ignore the fact that the reduction produces non-submodular terms leading to potentially many pixels unlabeled. Moreover, all the reduction techniques produce additional terms in the energy function and they grow exponentially with the maximum clique size. This consequently yields exponential growth of the time complexity as well as dimensionality. Message passing algorithms have also improved. To solve higher-order MRFs, belief propagation variants have been introduced. Lan *et al.* proposed some approximation methods for BP with learned potentials [49]. Potetz proposed a technique to compute belief propagation messages in time linear with the size of the largest clique for some class of energy functions [11, 7]. However, message passing algorithms applied to the higher-order MRF usually need much longer convergence time than Graph Cuts.

3.3 Proposed algorithm

The proposed algorithm is called Markov Chain Monte Carlo combined with General Deterministic algorithms (MCMC-GD). In this section, we explain the proposed MCMC-GD algorithm in detail. Our basic strategy is to combine deterministic algorithms in the structure of Pop-MCMC. As mentioned above, we can take advantages of the combination. However, the combination of sampling and deterministic algorithms is not an easy task. Careless embedment of deterministic algorithms in the sampling algorithm easily causes trouble.

The overall structure of MCMC-GD is described in Algorithm 2. The structure of MCMC-GD is similar to conventional population-based MCMC. The difference comes from the design of new kernel in the MCMC algorithm. This new kernel is proposed to make use of deterministic methods. It is based on the snooker crossover [37]. The detailed procedures for each step will be described in the following subsections.

3.3.1 Population-based sampling framework for MCMC-GD

To present overall structure of MCMC-GD, we would like to begin with MCMC first. MCMC algorithms have been used to sample from the target distribution $p(\mathbf{x})$. It generates a sequence of samples $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots$ using a Markov chain. A tth sample $\mathbf{x}^{(t)}$ is drawn from a conditional distribution $q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})$. We call $q(\cdot|\cdot)$ the kernel of the Markov chain. A kernel $q(\cdot|\cdot)$ is reversible if and only if

$$p(\mathbf{x}^{(t-1)})q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = p(\mathbf{x}^{(t)})q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}). \tag{3.2}$$

This is also called detailed balance condition. If a kernel $q(\cdot|\cdot)$ satisfies detailed balance, the Markov chain process $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots$ generated by the kernel converges to the target distribution $p(\mathbf{x})$.

Along with simulated annealing, MCMC has also been used to obtain an optimum sample of the target function, *i.e.* a sample $\mathbf{x}$ which maximizes $p(\mathbf{x})$. In MCMC, a new sample is drawn from the previous sample with a local transition probability, based on the Markov chain. Since most MCMC algorithms allow only local moves, in a large solution space it takes a very long time to reach the global optimum. To overcome the limitations of MCMC, Pop-MCMC has recently been applied to the vision problem [27, 28].

Pop-MCMC or evolutionary Monte Carlo is a stochastic simulation algorithm that combines a population of Metropolis–Hastings samplers and Evolutionary Algorithms to improve the performance of MCMC samplers. Pop-MCMC generates multiple chains in parallel. Each chain has a different target distribution $p_i(\mathbf{x}) = (1/Z_i)\{p(\mathbf{x})\}^{1/T_i}$ where Z_i is a constant to make the integral of the function equal to one and T_i represents the temperature for the ith chain. From multiple chains, multiple samples are drawn at the same time and they exchange information with each other. This enables global moves of samples which consequently make the mixing rate of drawn samples faster. In terms of optimization, the fast mixing rate means fast convergence to the global optimum.

Basically, MCMC-GD algorithm is built on the framework of Pop-MCMC. We first build the target distribution using Eq 3.3.

$$p(\mathbf{x}) = \frac{1}{Z}\exp\{-E(\mathbf{x})\}, \tag{3.3}$$

where Z is a constant to make the integral of the distribution function equal to one.

Note that, the domain of the target distribution $p(\cdot)$ is a real-valued space while that of the energy function $E(\cdot)$ is a integer-valued space. With the target distribution $p(\mathbf{x})$, we construct multiple chains with probability distribution of chain i as

$$p_i(\mathbf{x}) = (1/Z_i)\{p(\mathbf{x})\}^{1/T_i} \tag{3.4}$$

where Z_i is a normalizing constant and T_i is the temperature of the ith chain. In the chain with high temperature, the target distribution is nearly flat, where the heights of barriers between local optima are very low. Therefore, the samples in such chains can freely wander in contrast to the samples in a chain with low temperature. By exchanging these higher-temperature configurations with the configuration of a low temperature of our interest, we can allow the low temperature simulation to sample configurations much more efficiently than with local Metropolis updates only. This leads to a faster mixing rate between samples, and helps escape from local minima. The appropriate sequence of the temperatures depends on the given energy function. It is empirically determined (see Section 3.4 for discussions on how it is determined). Note that p_i is defined on real number space.

Given an original target distribution $p(\mathbf{x})$, a new expanded target distribution is defined as follows:

$$p^*(\mathbf{x}_{1:N}) = \prod_{i=1}^{N} p_i(\mathbf{x}_i), \tag{3.5}$$

where N is the number of chains to use. $\mathbf{x}_{1:N} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ is a population composed of samples of N chains. Our new goal is to generate a sequence of the population of samples $\mathbf{x}_{1:N}^{(0)}, \mathbf{x}_{1:N}^{(1)}, \cdots$ using a kernel $q(\mathbf{x}_{1:N}^{(t)}|\mathbf{x}_{1:N}^{(t-1)})$.

In this population-based sampling framework, deterministic algorithms are combined by using a new MCMC kernel. If we simply apply deterministic algorithms as the kernel of the sampling algorithm, it might violate the reversibility condition of the MCMC. Consequently, it is impossible to satisfy detailed balance. That is, we are not able to sample from the target probability function. Next subsection describes the design of kernel by which the deterministic algorithms are successfully combined with the sampling algorithm.

3.3.2 Kernel design

In this subsection, we present a new MCMC kernel design which enables the combination between sampling and deterministic method while satisfying reversibility and detailed balance. This new MCMC kernel generates a proposal sample by using deterministic algorithms. It enables much better move than widely-used random

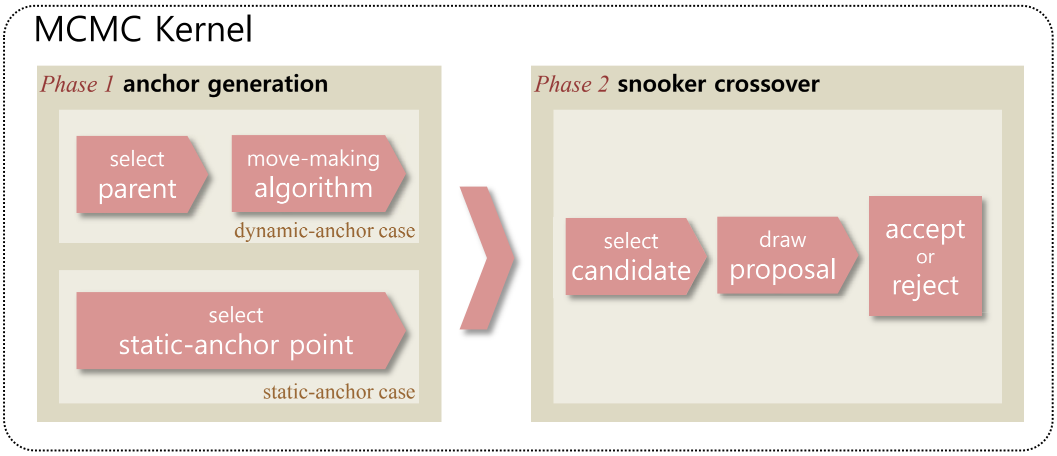


Figure 3.1: MCMC kernel for combining existing methods.

perturbation. The proposed kernel is composed of two phases: anchor generation and snooker crossover. Deterministic algorithms are employed in the first phase. The whole procedure for the kernel is illustrated in Figure 3.1.

3.3.2.1 Anchor generation

First phase is the anchor generation. The purpose of this phase is to generate anchors which have substantially low energy using the deterministic algorithms. In this subsection, we propose two different types of anchors according to the class of the algorithms to be combined: *Dynamic anchors* and *static anchors*. Although the dynamic anchor plays important role to produce well spread anchors, not every deterministic method can be used to generate the dynamic anchor. The Dynamic anchors are generated by using move-making algorithms. To also use other than move-making algorithms, such as message-passing algorithms, static anchors are proposed.

To generate dynamic anchors, we first select a sample $\mathbf{x}_p^{(t)}$ in the population.

We call this sample a parent. And then, we run a few iterations of a move-making algorithm with the parent as an initial. The resulting solution is used as the anchor for the next phase. This anchor is dynamically generated and destroyed while the algorithm is running.

To generate the static anchors, we initially run message-passing algorithms such as the TRW and BP before MCMC-GD starts. Those solutions are used as the anchor points. While the algorithm is running, we just pick one of the static anchors with uniform probability.

At each iteration, a single anchor is obtained by either dynamic or static anchor generation. To choose the type of the anchor, we draw a uniform random number U uniformly from interval $[0, 1)$ as mentioned in Algorithm 2. The random number U is compared with predefined dynamic anchor-based proposal rate Q_D which controls the relative weight of the dynamic and static anchor-based proposals. According to the value of U, we choose either the dynamic or static anchor-based proposals as the next proposal.

For the extreme cases, we can set $Q_D = 1$ and $Q_D = 0$. When Q_D is set to 1, the algorithm only uses dynamic anchors. On the other hand, when Q_D is 0, the algorithm only uses static anchors. Those algorithms are called MCMC-D and MCMC-S, respectively. (D and S stand for Dynamic and Static.) The relative performance of those variations is reported in experimental section.

3.3.2.2 Snooker crossover

After an anchor is chosen, we perform sampling by using snooker crossover (Figure 3.2). The snooker crossover has been proposed by Liang and Wong [37] for the MCMC sampling in a real-valued space. It start from selecting a sample $\mathbf{x}_c^{(t)}$ other

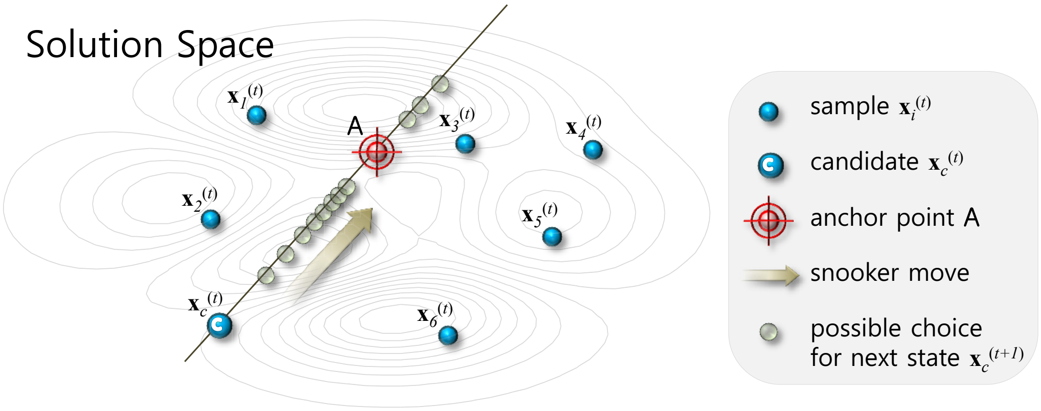


Figure 3.2: Snooker crossover. A cadidate sample $\mathbf{x}_c^{(t)}$ is randomly selected from the population $\{\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_N^{(t)}\}$. The sample $\mathbf{x}_c^{(t)}$ is then updated with a newly proposed sample $\mathbf{x}_c^{(t+1)}$ which is generated by the line sampling along the direction passing through the candidate $\mathbf{x}_c^{(t)}$ and the anchor point $\mathcal{A}$.

than the parent from the population $\{\mathbf{x}_1^{(t)}, \cdots, \mathbf{x}_N^{(t)}\}$. This sample is called a candidate. After the candidate is chosen, we perform snooker crossover with the anchor point $\mathcal{A}$. In the conventional snooker crossover, the anchor point is set to be the one of the samples chosen from population. On the other hand, in our algorithm, the anchor point comes from the result of the deterministic methods. A newly-generated sample $\mathbf{x}_c^{(t+1)}$ lies on the line going through the anchor and the candidate according to:

$$\mathbf{x}_c^{(t+1)} = \mathbf{x}_c^{(t)} \kappa \exp(s) + \mathcal{A}(1 - \kappa \exp(s)), \tag{3.6}$$

where s and κ are control parameters of snooker crossover. s is the random variable taken from the predefined set S with probability distribution $r(s)$. The set S can be designed as any set closed under the operator $\bar{s}$, which is defined by $-s$. The parameter s controls the distance between the newly-generated sample and the anchor. Small s results in the new sample being close to the anchor and large s results in the new sample being far away from the anchor. κ can be fixed either by $+1$ and -1 or randomly chosen among $+1$ and -1 with equal probability. κ will decide if the new sample, started from the candidate, passes over the anchor or not. When κ is -1, the newly-generated sample lies on the ray from the anchor in the opposite direction to the candidate. That is, the new sample passes over the anchor point. When κ is $+1$, the newly-generated sample lies on the ray from the anchor through the candidate.

After snooker crossover, the candidate $\mathbf{x}_c^{(t)}$ is substituted with the new sample $\mathbf{x}_c^{(t+1)}$ according to the Metropolis–Hastings rule with the acceptance probability:

$$\alpha = \min(1, \gamma), \tag{3.7}$$

where

$$\begin{aligned}
\gamma &= \frac{p_c(\mathbf{x}_c^{(t+1)})p_p(\mathbf{x}_p^{(t+1)})}{p_c(\mathbf{x}_c^{(t)})p_p(\mathbf{x}_p^{(t)})} \cdot \frac{q(\mathbf{x}_c^{(t)}, \mathbf{x}_p^{(t)} | \mathbf{x}_c^{(t+1)}, \mathbf{x}_p^{(t+1)})}{q(\mathbf{x}_c^{(t+1)}, \mathbf{x}_p^{(t+1)} | \mathbf{x}_c^{(t)}, \mathbf{x}_p^{(t)})} \\
&= \frac{p_c(\mathbf{x}_c^{(t+1)})}{p_c(\mathbf{x}_c^{(t)})} \cdot \frac{q(\mathbf{x}_c^{(t)} | \mathbf{x}_c^{(t+1)}, \mathbf{x}_p^{(t)})}{q(\mathbf{x}_c^{(t+1)} | \mathbf{x}_c^{(t)}, \mathbf{x}_p^{(t)})} \\
&= \frac{p_c(\mathbf{x}_c^{(t+1)})}{p_c(\mathbf{x}_c^{(t)})} \cdot \frac{r(-s)}{r(s)} \\
&= \exp\left[\frac{E(\mathbf{x}_c^{(t)}) - E(\mathbf{x}_c^{(t+1)})}{T_c}\right] \cdot \frac{r(-s)}{r(s)}.
\end{aligned} \tag{3.8}$$

Note that the reverse transition is attained by selecting $-s$ from S.

Instead of a single candidate, we can also pick multiple candidates at each iteration. The opposite extreme of using a single candidate is to take all the samples as candidates except the parent.

3.4 Experiments

3.4.1 Analysis on synthetic MRF problems

3.4.1.1 Pairwise MRF problems

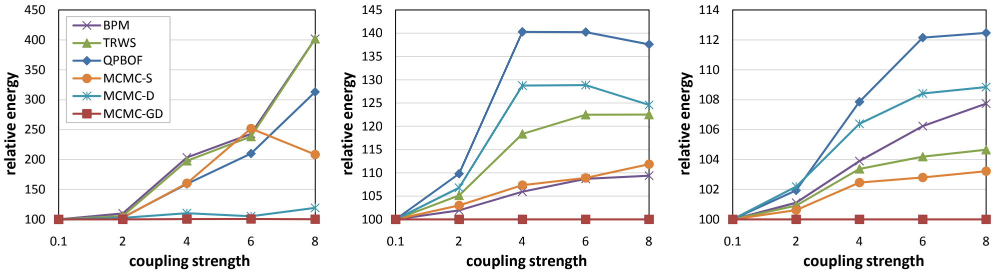
In this subsection, we analyze the performance of the proposed MCMC-GD algorithm while varying the difficulty of the target energy functions. To this end, the synthetic MRF problems were used so that the difficulty of the problem can be easily controlled. The difficulty of the MRF problems depend on many factors: the ratio of non-submodular terms, the coupling strength, the graph size, the number of labels, the connectivity, etc.

For graph construction, we followed the synthetic MRF construction in Komodakis's work [19]. We built multi-label MRFs defined on N by N grid graphs with four-neighborhood structures. We set the unary term of each node with a randomly generated number from Gaussian distribution $\mathcal{N}(0,1)$. The pairwise terms were set as:

$$\theta_{st}(x_s, x_t) = \begin{cases} 0 & \text{if } x_s = x_t, \\ \lambda_{st} & \text{if } x_s \neq x_t, \end{cases} \tag{3.9}$$

where λ_{st} was drawn from $|\mathcal{N}(0,\sigma^2)|$ for submodular terms and from $-|\mathcal{N}(0,\sigma^2)|$ for non-submodular terms. The parameter ρ controls the percentage of non-submodular terms and the parameter σ controls the coupling strength.

For MCMC-GD, identical control parameters were used for all the experiment in this subsection. We used 100 chains and the temperature of ith chain was set to i. For the snooker crossover, κ was randomly chosen among $+1$ and -1 with



(a) $\rho = 1\%$ (b) $\rho = 25\%$ (c) $\rho = 50\%$

Figure 3.3: Experimental results on the synthetic MRF problems. 30 by 30 grid graphs are generated. The difficulty of the energy function defined on the graph is controlled by two paramters: the percentage of non-submodular terms ρ and the coupling strength σ. As the problem becomes harder to solve, the gap in the performance between the proposed algorithm and others is getting larger.

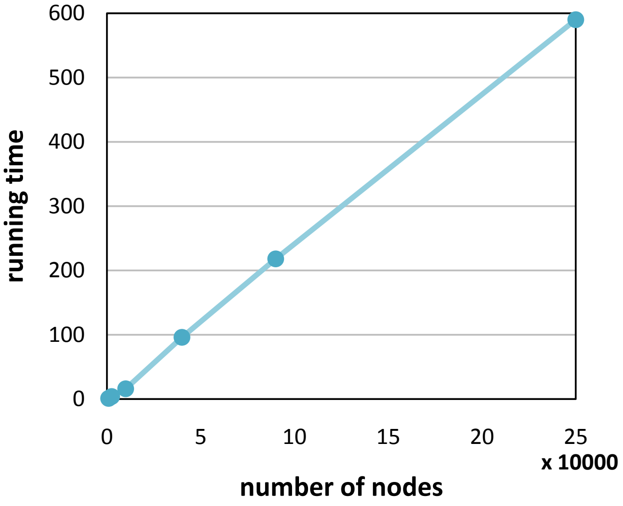
equal probability. $\exp(s)$ was drawn from $\{0.1, 0.5, 2, 10\}$ with probability 0.5, 0.2, 0.2, and 0.1, respectively. Q_D was set to 0.9. The effect of the control parameters will be discussed in the later section in detail. At each iteration, we selected single candidate. For the dynamic anchor-based proposal, we used a single iteration of QPBO algorithm. The single iteration of QPBO algorithm is composed of a single α-expansion using QPBO with randomly chosen α-label. Because of non-submodular terms, QPBO leaves unlabeled pixels. Those unlabeled pixels were assigned to current labels. For the static proposal, static anchors were obtained using TRW-S and BP-S. Please note that the time for generating static anchor is excluded from the running time except for the Section 3.4.3.

There are two issues in MCMC-GD. First issue is how to assign appropriate values for various control parameters. The effect of choices of different parameters will be discussed in the later subsection. Second issue is to determine algorithms for the anchor generation. We believe it would be hard to provide rigorous mathematical theories in this case. Instead, we can provide rough guideline to obtain better solutions. We need to understand what are good anchors. First, good anchors have low energy. Second, good anchors need to be well-distributed. Therefore, it is recommended to use available state-of-the-art algorithms for anchor generation. In addition, we suggest using various algorithms which have different search schemes.

On the other hand, it is left up to users to make a choice of algorithms to use. We propose a general framework for combining different algorithms. In this framework, any existing algorithms can be combined together. We can provide only a rough guideline. We experimentally found that it is usually better to combine good-performing algorithms and to combine various algorithms. It is, however, not good idea to combine too many algorithms. Our recommendation is to use one graph cuts-based algorithm and one message passing-based algorithm if both are available.

In the first set experiments, we compared the performance of MCMC-GD and other algorithms while varying the ratio of non-submodular terms and the coupling strength to control the difficulty of the problems. It has been shown that the amount of non-submodular terms affects the performance of QPBO by Rother *et al.* [15]. Our experimental results show that the performance of other well-known algorithms such as TRW and BP also depend on the amount of non-submodular terms. It consequently tells the difficulty of the MRF problems depends on the amount of the non-submodular terms. The coupling strength refers to the relative strength of pairwise versus unary terms. It is known that the MRF problems become more difficult as the coupling strength is getting larger [15, 19]. For the comparison, we applied QPBO, TRW-S, and two different variants of BP: BP-S [2] and BP-M [16]. We also applied MCMC-D and MCMC-S which are the variants of MCMC-GD.

The parameters ρ was set to 1%, 25%, and 50% and the coupling strength σ was



Graph Size	Time
30 x 30	1 sec
50 x 50	4 sec
100 x 100	16 sec
200 x 200	96 sec
300 x 300	218 sec
500 x 500	590 sec

Figure 3.4: The running time of MCMC-GD applied to the synthetic problems with different number of nodes. The number of nodes was set to 30 by 30, 50 by 50, 100 by 100, 200 by 200, 300 by 300, and 500 by 500. The parameter ρ and the coupling strength were fixed to 50% and 8, respectively. The running time increases almost linearly with the number of nodes.

set to 0.1, 2, 4, 6, and 8. On the other hand, the number of labels and the size of the graphs are fixed to be 5 and 30 by 30, respectively. For each parameter setting, we construct 20 different instances of MRF problem. For each instance, final energies are normalized so that the minimum energy is to be 100 and the average of the final energies was obtained.

The results of the first set of experiments are summarized in Figure 3.3. We applied MCMC-GD, MCMC-D, MCMC-S, QPBO, TRW-S, BP-S, and BP-M. In the graph, the x-axis represents the coupling strength and the y-axis are relative energy given by $100\times(energy\quad of solution)/(minimum\quad energy)\%$. We obtained better results by combination. Note that MCMC-GD always obtained the lowest energy among all other methods. Moreover, the energy gap between MCMC-GD and others became larger as the problem was more difficult in terms of the ratio of non-submodular terms and the coupling strength. The running time of MCMC-GD algorithm was set to 8 s. QPBO is the fastest among all the methods. It took less than 0.1 s to converge. For TRW-S, BP-S, and BP-M, the maximum number of iterations was set to 2000. It took 3–5 s for TRW-S to terminate and 7–9 s for BP-S and BP-M to terminate.

In the second set experiments, we analyzed the complexity and the performance of MCMC-GD for different graph sizes. To measure time complexity, we need to set up reasonable stopping criteria. It is not easy to decide when the algorithm should be terminated. When we run population-based stochastic optimization algorithms for a long enough time (*e.g.*, an exponentially long time), many of them will finally end up with the globally optimal solution. This, however, is waste of computational resources since at the early phase of the algorithm the energy is severely decreased whereas the decrease is too small at the late phase. Consequently, algorithms need to be terminated with an appropriate stopping criterion. One of the best ways is to terminate the algorithm when the discrepancy between the global minimum energy and the current energy is small enough. In practical cases, however, the knowledge about global optima is usually not available. In this experiment, instead, we first execute the algorithm for long enough time to get reasonably low energy solutions. After that, we measured the discrepancy between the energy of those solutions and the current energy. The algorithm is terminated when the discrepancy becomes less than 0.5%.

The running time is plotted on Figure 3.4. The number of nodes was set to 30 by 30, 50 by 50, 100 by 100, 200 by 200, 300 by 300, and 500 by 500. The parameter ρ, the coupling strength, and the number of labels were fixed to 50%, 8, and 5, respectively.

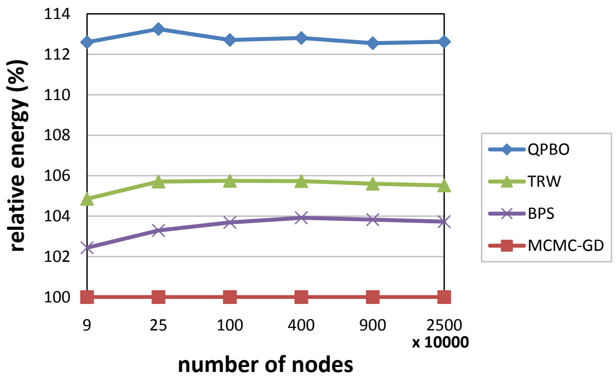


Figure 3.5: Relative energies of algorithms applied to the synthetic problems with different number of nodes. The number of nodes was set to 30 by 30, 50 by 50, 100 by 100, 200 by 200, 300 by 300, and 500 by 500. The parameter ρ and the coupling strength were fixed to 50% and 8, respectively. MCMC-GD always found lowest energy solution among all applied algorithm. Note that relative energy of each algorithm is almost similar without regard to the number of nodes.

For each parameter setting, we construct 20 different instances of MRF problem. For each instance, final energies are normalized so that the minimum energy is to be 100 and the average of the final energies was obtained. The parameters for MCMC-GD algorithms were set to the same as the first experiment. The discrepancy of the energy was measured at every second. For 30 by 30 graph the algorithm converged in one second and for 500 by 500 graph the algorithm converged in 590 s. The detailed results are summarized in Figure 3.4. The running time increases almost linearly with the number of nodes. In Figure 3.5, relative energies of algorithms applied to the synthetic problems with different number of nodes are shown. MCMC-GD always found lowest energy solution without regard to the graph size. The ranking and the relative energies of algorithms remain almost same across different graph sizes.

Figure 3.5 shows the comparison of the final energy values obtained by different

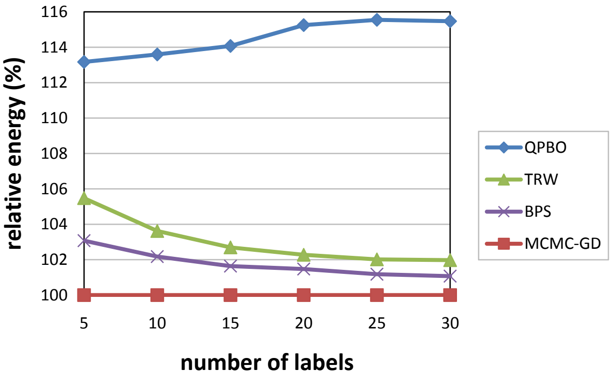


Figure 3.6: Relative energies of algorithms applied to the synthetic problems with different number of labels. The number of labels was set to 5, 10, 15, 20, 25, and 30. The parameter ρ and the coupling strength were fixed to 50% and 8, respectively. MCMC-GD always found lowest energy solution among all applied algorithm. BP-S and TRW is getting better when the number of labels become larger.

algorithms while varying the number of nodes. MCMC-GD, QPBO, TRW, BP-S is used for the experiments. MCMC-GD always obtained the lowest energy solution among the tested algorithms. The performance between algorithms remains relatively unchanged with increase in the number of nodes. It suggests that the graph size does not have significant influence on the relative performance of the algorithms.

In the third set experiments, we compared the performance of MCMC-GD for different number of labels. The number of labels was set to 5, 10, 15, 20, 25, and 30. The parameter ρ, the coupling strength, and the size of the graph were fixed to 50%, 8, and 30 by 30, respectively. For each parameter setting, we construct 20 different instances of MRF problem. For each instance, final energies are normalized so that the minimum energy is to be 100 and the average of the final energies was obtained.

Figure 3.6 shows the comparison of the final energy values obtained by different algorithms while varying the number of labels. MCMC-GD, QPBO, TRW, BP-S is

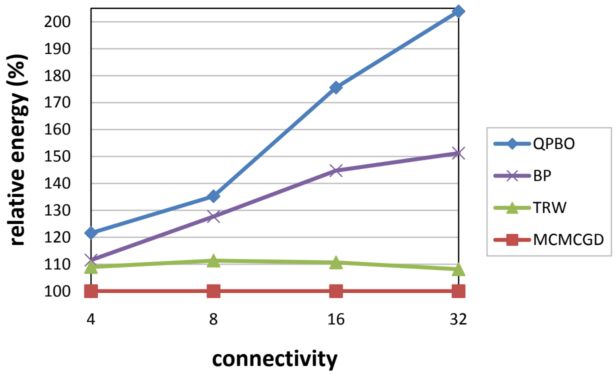


Figure 3.7: Relative energies of algorithms applied to the synthetic problems with different connectivity. The number of neighbors for each node was set to 4, 8, 16, and 32. The parameter ρ and the coupling strength were fixed to 50% and 8, respectively. MCMC-GD always found lowest energy solution among all applied algorithm. BP-S and TRW is getting better when the number of labels become larger.

used for the experiments. MCMC-GD always obtained the lowest energy solution among the tested algorithms. As the number of labels gets larger, the performance of QPBO was degraded. On the other hand, the performance of TRW and BP-S was enhanced when the number of labels gets larger.

In the fourth set experiments, we compared the performance of MCMC-GD while varying the number of neighbors (*i.e.* connectivity). Contrast to previous experiments, we did not used grid graph to easily control the connectivity of graphs. The number of graph was set to 33 and the neighborhood structure is following. For the neighborhood structure, we assign the nodes $i-k, \cdots, i-1, i+1, \cdots, i+k$ as neighbors of the nodes i. In this case, the connectivity of graph is $2k$. In case connectivity is 32, the graph is fully connected. The connectivity was set to 4, 8, 16, and 32. The parameter ρ, the coupling strength, the number of graphs, and the size of the graph were fixed to 50%, 8, 5, and 30 by 30, respectively. For each parameter setting, experiments were repeated 20 times with different random number seeds to generate 20 different target functions and the average of the final energies was obtained.

Figure 3.7 shows the comparison of the final energy values obtained by different algorithms while varying the connectivity of graphs. MCMC-GD, QPBO, TRW, BP-S is used for the experiments. MCMC-GD always obtained the lowest energy solution among the tested algorithms. As the connectivity gets larger, the performance of QPBO and BP was degraded. On the other hand, the performance of MCMC-GD and TRW remained relatively unchanged.

3.4.1.2 Parameter analysis

In this subsection, we analyze the performance of the proposed MCMC-GD algorithm while varying the parameters used in MCMC-GD. The parameters include Q_D, which is the ratio of the dynamic and static anchor-based proposals, the temperatures, and the number of chains (*i.e.* the population size). We are going to examine how stable MCMC-GD is with respect to the selection of the parameters.

Throughout the experiments in this subsection, the algorithm is tested on the same set of energy functions. The energy is defined on 30 by 30 grid graph. The unary and pairwise terms are defined the same way as in the previous subsection. The parameters for the graph were fixed as follows: The ratio of non-submodular term ρ was set to 50%; the coupling strength was set to 8; and the number of the graph was fixed to 5. We construct 20 different instances of MRF problem. For each instance, final energies are normalized so that the minimum energy is to be 100 and the average of the final energies was obtained.

In the first set of experiments, we analyzed the performance of MCMC-GD while

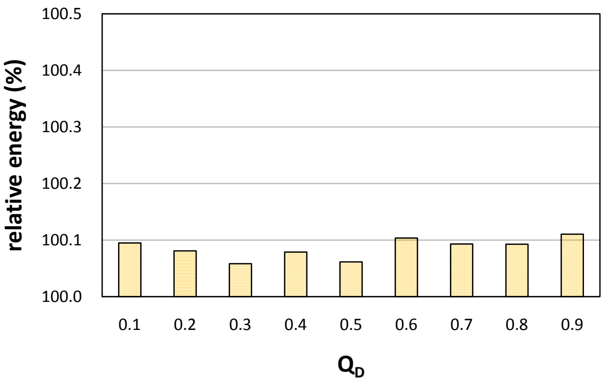


Figure 3.8: Sensitivity of the proposed algorithm w.r.t. to parameter Q_D. Experiments were performed on the synthetic MRF problems. The parameter ρ and the coupling strength were set to 50% and 8. The algorithm is stable to change of the parameter Q_D.

varying the parameter Q_D. The parameter Q_D controls the relative weight of the dynamic and static anchor-based proposals. When the Q_D gets larger, the algorithm uses more dynamic anchor-based proposals, and vice versa. The other parameters and settings were fixed to the same as in previous subsection.

Figure 3.8 shows the comparison of the final energy values obtained by MCMC-GD while varying the parameter Q_D. The parameter Q_D was changed from 0.1 to 0.9 by 0.1. In every case, the final relative energy was less than 100.2%. It shows that MCMC-GD is stable with choices of the parameter Q_D as long as it is not chosen from 0 or 1.

In the second set of experiments, we analyzed the performance of MCMC-GD while varying the temperature for each chain. Throughout all the experiments, only the maximum and minimum temperatures are defined. In-between values are defined by linear interpolation of them. The maximum temperature was varied from 1 to $10,000$ and the minimum temperature was varied from 0.1 to 1000. The other

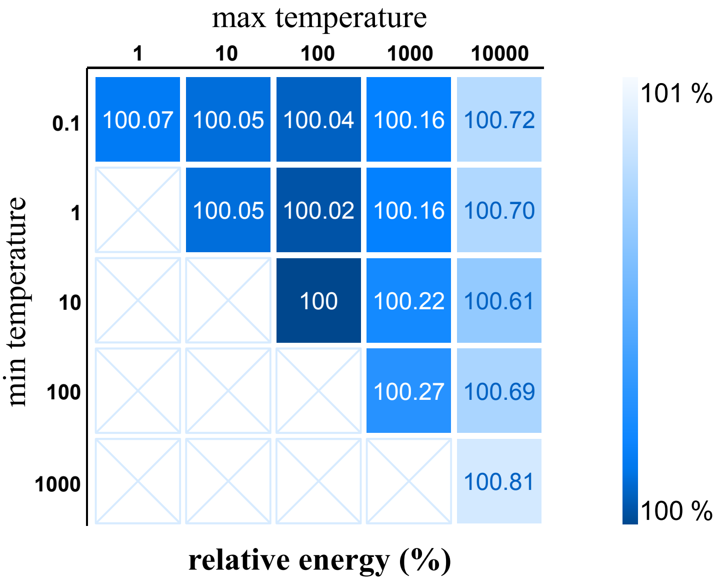


Figure 3.9: Algorithm robustness to the temperature setting. Experiments were performed on the synthetic MRF problems. The parameter ρ was set to 50% and the coupling strength was set to 8. The algorithm is stable to change of the temperatures.

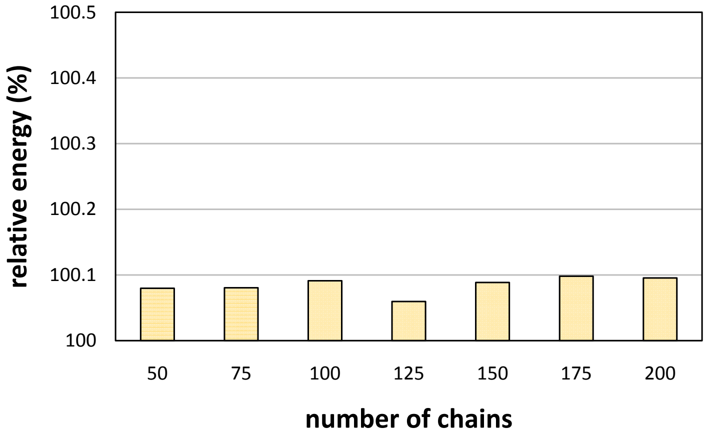
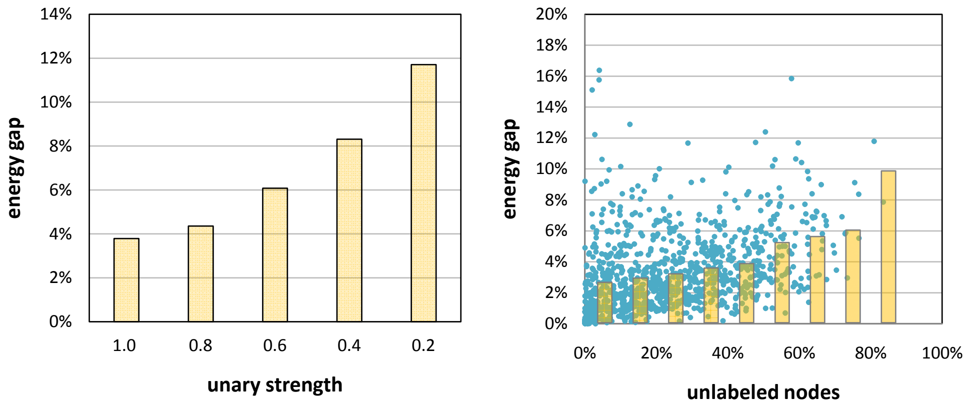


Figure 3.10: Algorithm robustness to the number of chains. Experiments were performed on the synthetic MRF problems. The parameter ρ was set to 50% and the coupling strength was set to 8. The algorithm is stable to change of the number of chains.

parameters and settings were fixed to the same as in previous subsection.

Figure 3.9 shows the comparison of the final energy values obtained by MCMC-GD while varying the temperatures of chains. In every case, the final relative energy was less than 101%. It shows that MCMC-GD is stable with choices of the temperature as long as it is selected from a reasonable range.

In the third set of experiments, we analyzed the performance of MCMC-GD while varying the number of chains. Figure 3.10 shows the comparison of the final energy values obtained by MCMC-GD while varying the number of chains. In every case, the final relative energy was less than 100.1%. It shows that MCMC-GD is stable with choices of the number of chains as long as it is selected from a reasonable range.



(a) (b)

Figure 3.11: The energy gaps between QPBO and MCMC-GD against the difficulties of the problems. The energy gaps are given by (*energy of QPBO*) − (*energy of MCMC-GD*). Combined with QPBO, MCMC-GD always obtains better solutions than QPBO does. Especially, energy gap between QPBO and MCMC-GD is larger when the problems are more difficult. Difficulties are (a) controlled by changing unary strength and (b) estimated by amount of unlabeled nodes in QPBO. In (b), a single blue dot represents each individual experiment and the histogram of the blue dots is drawn by yellow bars.

3.4.1.3 Higher-order MRF problems

We also evaluated the proposed algorithm on the higher-order MRF synthetic problems. For the experiments, we constructed 10 by 10 grid graphs with unary and higher-order potentials. The labels were discrete values between 0 and 255. To define unary terms, random values r_s between 0 and 255 are first assigned for each node s. Unary potential was defined by Normal distribution functions which have randomly assigned values as mean values and 20 as standard deviation:

$$\theta_s(x_s) \propto \exp\left\{-\frac{(x_s - r_s)^2}{2 \cdot 20^2}\right\}. \tag{3.10}$$

Higher-order clique potentials were defined by Fields of Experts (FoE) [3]:

$$\theta_c(\mathbf{x}_c) \propto \prod_{i=1}^{K} \left\{ 1 + \frac{1}{2}(J_i \cdot \mathbf{x}_c)^2 \right\}^{-\alpha_i}, \tag{3.11}$$

where J_i is an $n \times n$ linear filter, K is the number of filters, and α_i is a positive value. The parameters J_i and α_i are learned from a database of natural images. Instead of learning, however, we randomly generated parameters for the experiments here. Each element of the filter J_i and the parameter α were drawn from the uniform distribution in the interval $(-1, 1]$ and $(0, 2]$, respectively. The number of the filters was set to three.

Table 3.1: Average energies of the solutions over 1000 instances of synthetic problems.

	MCMC-GD	QPBO	BP
Energy	1737.3	1797.3	2283.5

First experiment of the higher-order MRFs shows how the performance changes against the unary strength. We applied QPBO [8] and MCMC-GD to randomly generated MRF problems while changing the unary strength. The unary strength was set to $0.2, 0.4, 0.6, 0.8$, and 1.0. For each unary strength setting, we construct 20 different instances of MRF problem. For each instance, final energies are normalized so that the minimum energy is to be 100 and the average of the final energies was obtained. For MCMC-GD, we set the number of chains and the temperatures to the same values as in previous experiments. The random variable κ and $\exp(s)$ was also drawn from the same distribution as in previous ones. At each iteration, we selected all sample except the parent as candidate. In this experiment, we only use dynamic anchors by setting Q_D to 0. For the dynamic anchor generation, QPBO was used. The running time of MCMC-GD are set to 30 s.

The results of the first experiment are summarized in Figure 3.11(a). The x-axis is unary strength and the y-axis is energy gap (energy difference) between QPBO and MCMC-GD. Note that MCMC-GD always obtains the lower energy than QPBO. The energy gap is getting larger when the problems are more difficult.

The second experiment of the higher-order MRFs analyzes the affect of non-submodular terms which are inevitable when we reduce higher-order clique potentials into pairwise ones. However, it is impossible to control the amount of non-submodular terms by manipulating higher-order clique potentials. Instead, we carried out 1000 experiments on the randomly generated MRF problems. And then, we estimated the amount of the non-submodular terms by estimating the difficulties using the percentage of the unlabeled nodes in QPBO process. It is well known that the unlabeled nodes in QPBO can be used to measure the difficulties of the problems [15].

The results for second experiment are summarized in Table 3.1 and Figure 3.11(b). Table 3.1 contains the average energies of the solutions from each algorithm over 1000 instances. MCMC-GD obtains the lowest energy solutions. Figure 3.11(b) shows the results in more detail. Each green dot represents each instance of experiments. The x-axis is ratio of the unlabeled pixels in QPBO. The bigger it is, the harder problem is. The y-axis is energy gap (energy difference) between QPBO and MCMC-GD. We also depicted the average energy gaps for every 10% with yellow bars. It is shown that the energy gap between QPBO and MCMC-GD is getting larger as the problem becomes more difficult.

3.4.2 Results on real problems

3.4.2.1 Photomontage

We also applied MCMC-GD to a practical vision problem known as photomontage [50, 2]. The photomontage problem seamlessly stitches multiple number of photos. Given a set of input images $I_1, I_2, \cdots, I_L$, the goal is to output a merged image by copying colors from one of the input images per pixel. It usually begins with some user strokes as a hard constraint. For each image, a user make marks which are desired to appear in output image. With this hard constraint, the photomontage merges input images into a single output image. In this experiment, the energy model was set to the same as that in Szeliski *et al.*'s paper [2] (second benchmark in photomontage). We used five input images so that the number of labels was also five. We applied our MCMC-GD algorithm as well as other methods.

Among all the benchmark MRF problems in Szeliski *et al.*'s comparative study [2], the photomontage is considered as the most difficult problem due to the intrinsic property of the energy formulation. It is because the energy of the photomontage problem is dominated by the smoothness cost. As shown in the previous subsection, large coupling strength makes the problem more difficult. In addition, the function itself is non-submodular which consequently leads the truncation for α-Expansion method. We also empirically found that fewer user strokes and larger clutter in the image made the problem even harder.

Now, the settings for MCMC-GD algorithm are as follows. First of all, we used 100 chains and the temperature of ith chain was set to $i \times 100$. The temperature was set to be a little higher than the synthetic cases due to the size of problem. We experimentally found that higher temperature setting gives better results when the size of problems are getting larger although the algorithm is rather robust to the change of the temperature (note Section 3.4.1.2). At each iteration, we selected all sample except the parent as candidates. This is better than single candidate because single move of the move-making deterministic algorithm takes long time in this problem. For the dynamic anchor-based proposal, we used five iterations of α-Expansion algorithm. Note that the number of iterations was set to the same as the number of labels. For the static proposal, static anchors were obtained using TRW-S. The parameter settings for the snooker crossover were the same as used for the synthetic MRF problems in the previous subsection. For TRW-S, BP-S, and BP-M, the maximum number of iterations was set to 2000.

The input images are shown in Figure 3.12. User strokes are represented by the white pixels. An example of quantitative results is provided in Figure 3.13. Upper row shows the resulting photomontage image of the each algorithm, and the lower row exhibits the corresponding color-coded image according to the labeling. Figure 3.14 presents the comparative energy plots of all the test algorithms against running time in seconds. Note that MCMC-GD algorithm always reached the lowest energy state among all other methods. The preprocessing time for obtaining the static anchor was not counted on the graph.

3.4.2.2 Inpainting

We also applied MCMC-GD to image inpainting which was formulated as higher-order MRF model. The energy function for inpainting is difficult to minimize because it does not have unary terms. It is shown in previous section that the smaller unary strength makes the problems more difficult. Therefore, inpainting problem is an appropriate application to compare the performance of the algorithms.

Given original image, we first mark 70% of pixels as '*unknown*'. In Figures 3.16(a) and 3.17(a), unknown pixels are represented by red color. And then, the goal is to restore the intensity values for those pixels. As a prior, we used FoE model which is learned in Roth's work [3].

Now, the settings for MCMC-GD algorithm are as follows. We used 100 chains and the temperature of ith chain was set to i. The random variable κ and $\exp(s)$ was drawn from the same distribution in previous experiments. At each iteration, we selected all sample except the parent as candidate. By setting Q_D to 0, we only used dynamic anchors. For dynamic anchor generation, 10 iterations of QPBO were used.

Table 3.2: Energy and PSNR for four images using MCMC-GD and QPBO algorithms. Both algorithms are run for 3000 s.

test image	**MCMC-GD**		**QPBO [8]**	
	energy	*PSNR*	*energy*	*PSNR*
Berkeley001	19278	27.20	20115	26.51
Berkeley002	20213	28.11	20964	27.67
Berkeley003	19351	25.93	19884	25.76
Berkeley004	32288	24.61	33040	24.28

We experimented on four images in the Berkeley segmentation database [51]. Final energies and the PSNR from QPBO and MCMC-GD are listed in table 3.2. PSNR is the Peak Signal-to-Noise Ratio given by $20\log_{10}(255/\sigma)$ where σ is the standard deviation of the intensity difference between the solution and the ground truth images. MCMC-GD always found lower energy solutions with higher PSNR. Original images of experiments are shown in Figure 3.15. Both quantitative and qualitative results are shown in Figures 3.16 and 3.17. In Figures 3.16(a) and 3.17(a), bottom left is the enlarged result image of QPBO and bottom right is that of MCMC-GD. We can see MCMC-GD produced less artifacts and the result of it looks better especially near the boundaries. Figures 3.16(b) and 3.17(b) show the energy versus time graph. While QPBO converges fast and almost does not decrease the energy after about 200 s, MCMC-GD kept decreasing the energy of the solution much lower than QPBO. The ratio of marked pixels also affects the performance of algorithm. By marking more pixels, we can make the problem more difficult. It is because more marked pixels means bigger problem size and larger connectivity. We found out that when more pixels are marked, the difference in performance between MCMC-GD and QPBO became larger, and vice versa.

3.4.3 Alternative approach: parallel anchor generation

In the preceding experiments, we presented the experimental results while excluding the running time for obtaining the static anchors. To obtain the static anchors, message passing algorithms are run before the MCMC-GD algorithm starts. For a fair comparison, the running time for obtaining the static anchors should be included. It consequently makes the MCMC-GD algorithm less satisfactory in terms of the time complexity.

In this subsection, we propose an alternative approach which shortens the whole running time including the time for generating the static anchors. Instead of running the message passing algorithms before the MCMC-GD algorithm starts, we propose to run the message passing algorithms in parallel with MCMC-GD sampling. For example, we can alternately perform the single iteration of the message passing algorithm and the single iteration of the MCMC-GD sampling. In that case, the static anchor will not remain same but changed as algorithm runs. The static anchor-based proposals will be generated with unconverged solutions. This idea can be generalized by alternately running the N iterations of the message passing algorithm and M iterations of the MCMC-GD sampling.

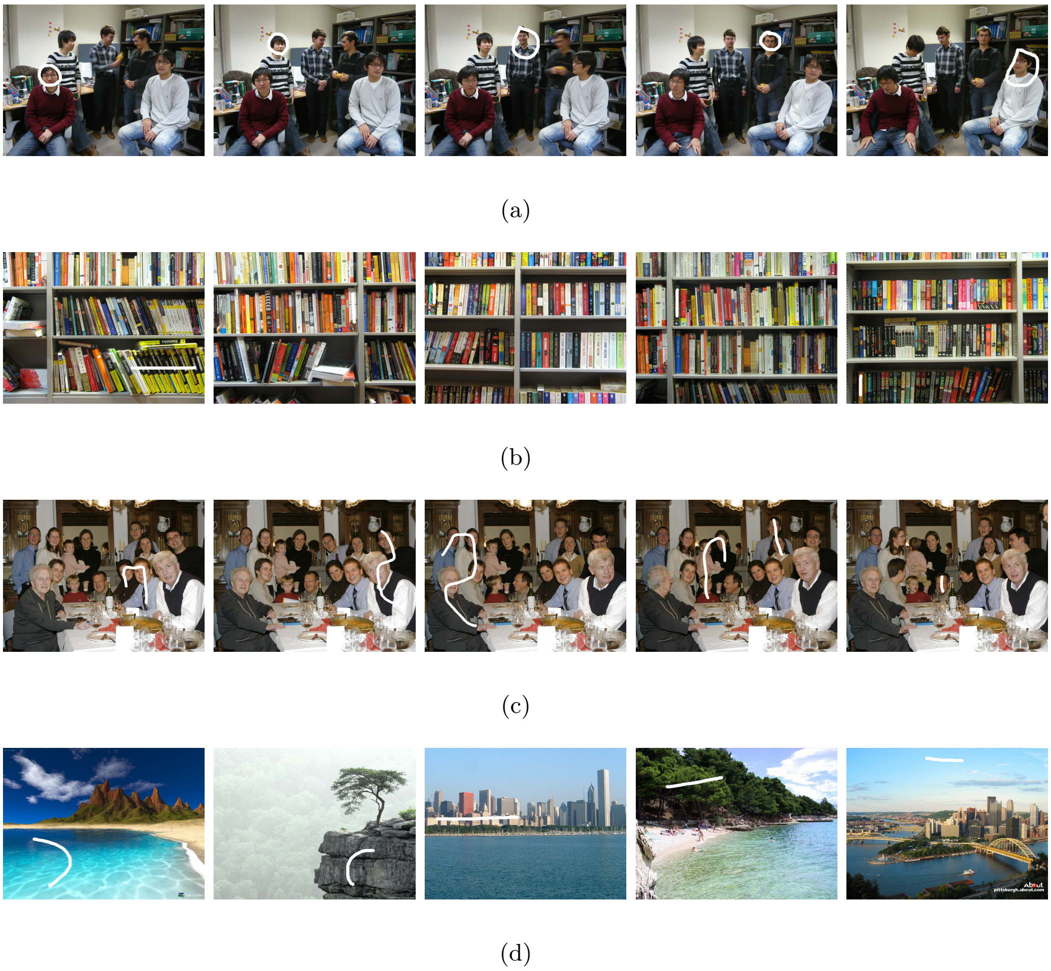
Figure 3.18 shows the comparison of the original MCMC-GD and its alternatives on Lab images. MCMC-GD(M:N) refers the variation of MCMC-GD, where M iterations of the message passing and N iterations of the sampling is alternatively performed. In the original MCMC-GD, the sampling procedure starts after 1000 s. In its variations, on the other hands, the sampling procedure starts at the beginning phases. It is shown that the alternative approach effectively reduce the total running time for MCMC-GD.

3.5 Summary

Although there have been great advances in solving simple MAP–MRF based vision problems, optimizing more complex MRF problems is still remained as challenging. Examples of the complex MRF problems include non-submodular energy functions, strongly coupled MRF, and high-order clique potentials. Most existing optimization algorithms have inherent limitations in solving those difficult problems. In this chapter, we proposed a new efficient algorithm called MCMC-GD that could cope with those difficult MRF problems. Basically, MCMC-GD is the sampling-based method (Pop-MCMC) combined with deterministic methods. By combination, the deterministic methods help the sampling-based method to rapidly move into the lower energy state. Moreover, the deterministic methods make the sampling-based method jump easily from one basin to another over the energy barrier. Consequently, the mixing rate was increased and we achieved faster convergence and better solutions. On the other hand, the sampling-based method helps deterministic methods not to be stuck in local minima. We experimentally showed that the proper combination of the different approaches could substantially improve the overall performance. Our new energy minimization framework will be useful in solving many challenging vision problems. Consequently, this will encourage the design of better yet more complex energy models for practical vision applications.

Algorithm 2 MCMC-GD algorithm

```
<Initialize>
Initialize the population $\mathbf{x}_{1:N}$
Set the temperatures $T_1 < T_2 < \cdots < T_N$
Run message passing algorithms to get solutions $\mathcal{A}^1_{static}, \mathcal{A}^2_{static}, \cdots, \mathcal{A}^K_{static}$
repeat
  <Anchor generation>
  if $U \sim [0,1] < Q_D$ then
    $p \sim \{1, 2, \cdots, N\}$
    $\mathcal{A} \leftarrow Move\_making\_algorithm(\mathbf{x}_p^{(t)})$
  else
    $k \sim \{1, 2, \cdots, K\}$
    $\mathcal{A} \leftarrow \mathcal{A}^K_{static}$
  end if
  <Snooker crossover>
  $c \sim \{1, 2, \cdots, N\} - \{p\}$
  $\kappa \sim \{+1, -1\}$
  $s \sim S$
  $\mathbf{x}_c^{(t+1)} \leftarrow \mathbf{x}_c^{(t)} \cdot \kappa \exp(s) + \mathcal{A} \cdot (1 - \kappa \exp(s))$
  Determine whether accept the new population or not by the Metropolis–Hastings rule.
until The algorithm converges.
```



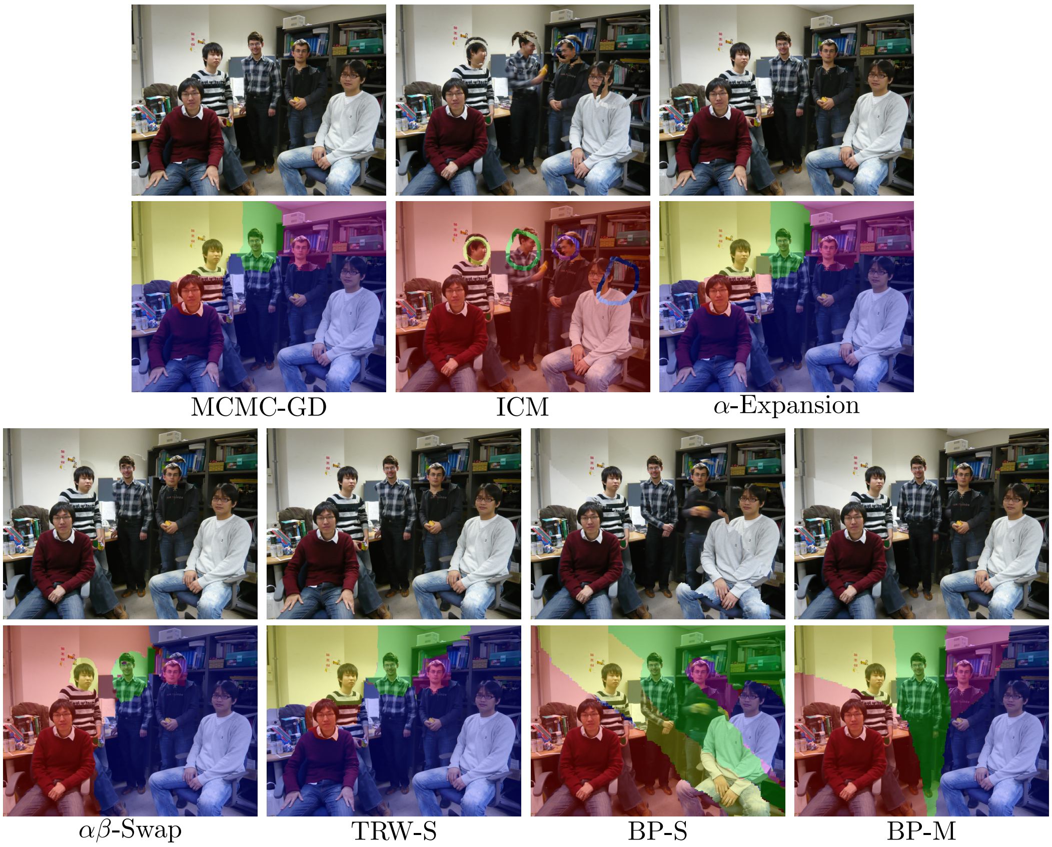
(a)

(b)

(c)

(d)

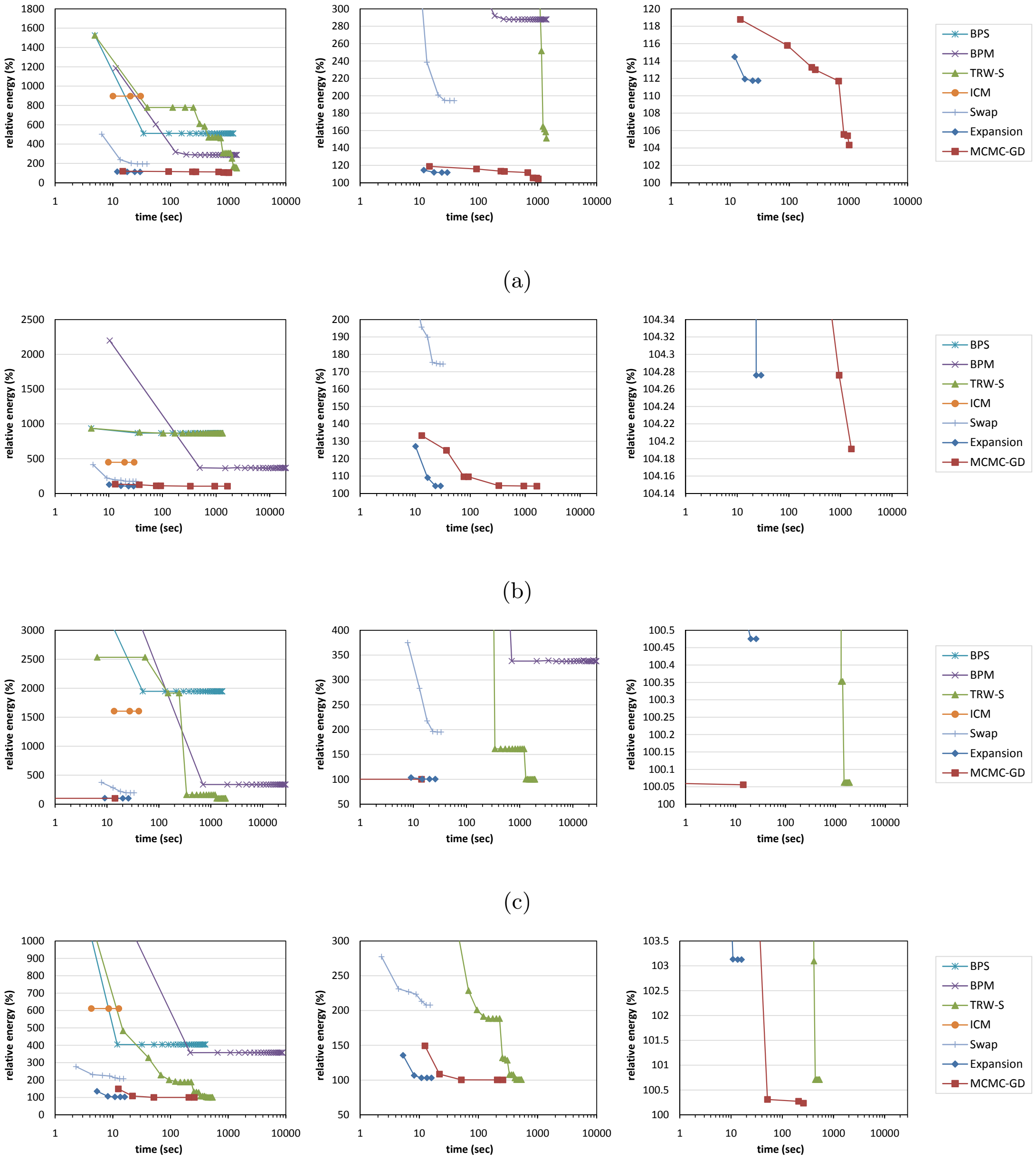
Figure 3.12: Input images of photomontage. (a) Lab images, (b) bookshelf images, (c) family images, and (d) landscape images.



MCMC-GD ICM α-Expansion

$\alpha\beta$-Swap TRW-S BP-S BP-M

Figure 3.13: Photomontage results. There is no single input image in which everybody is looking at the camera. The goal is to generate a merged image with five front-view faces. First and third rows show the resulting photomontage images of the each algorithm. Second and forth rows represents the corresponding color-coded image according to the labeling. Although energy function enforces seams visually unnoticeable, ICM, $\alpha\beta$-Swap, BP-S, and BP-M produce distinct seams. On the other hand, MCMC-GD, α-Expansion, and TRW-S give qualitatively good results.



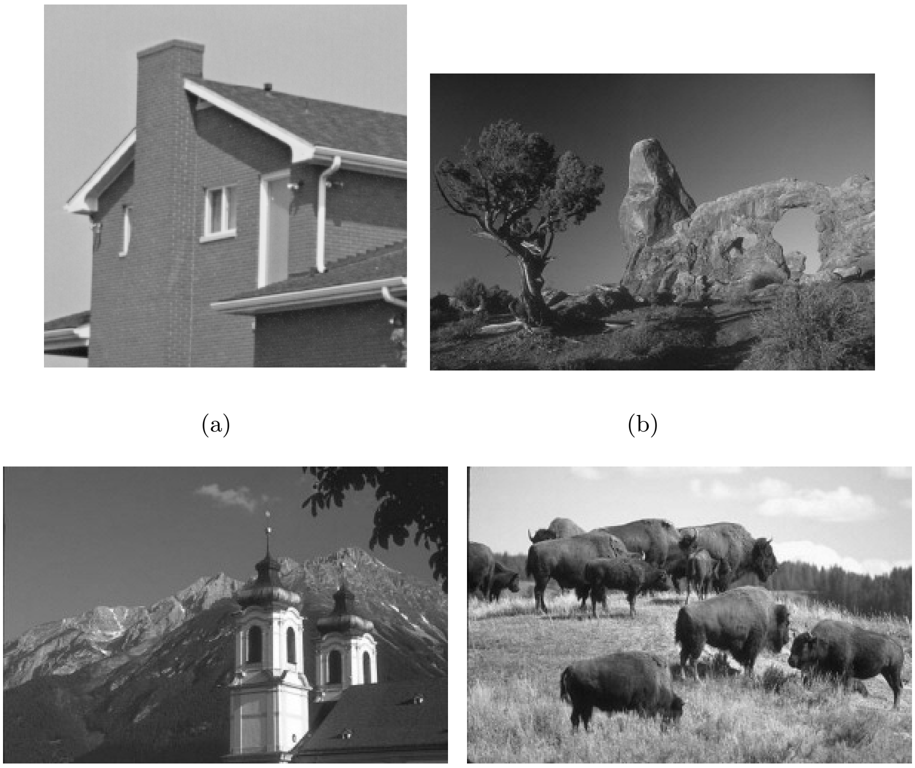
(a)

(b)

(c)

(d)

Figure 3.14: Experimental results on the photomontage problem: (a) Lab images, (b) bookshelf images, (c) family images, and (d) landscape images. Each result is depicted three times with increasing scale from left to right. In all cases, MCMC-GD found the solution with the lowest energy.



(a)

(b)

(c)

(d)

Figure 3.15: Input images of inpainting. (a) Berkeley001, (b) Berkeley002, (c) Berkeley003, and (d) Berkeley004.

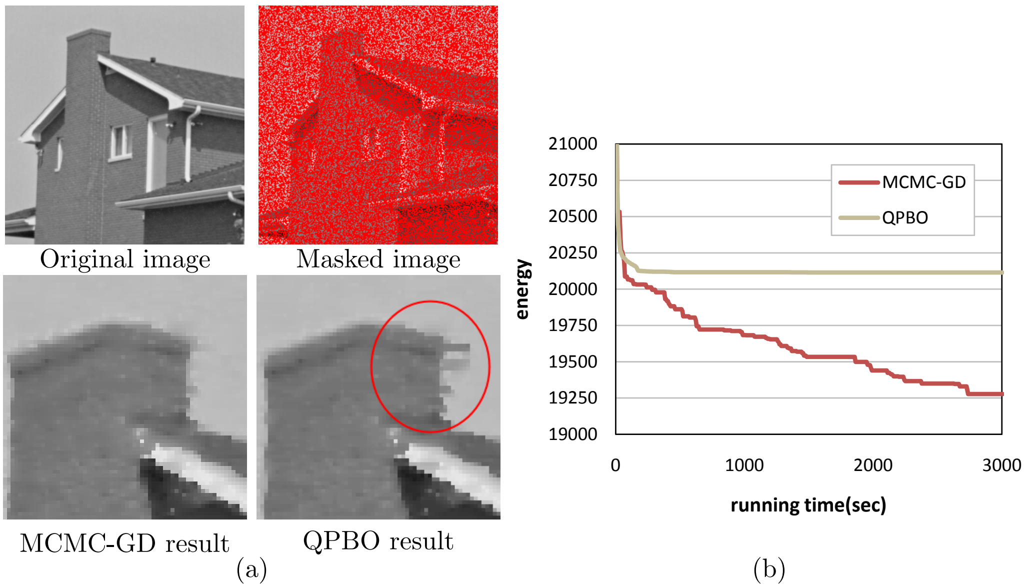


Figure 3.16: The inpainting results of the Berkeley001 image. (a) Qualitative and (b) quantitative comparisons of QPBO and MCMC-GD.

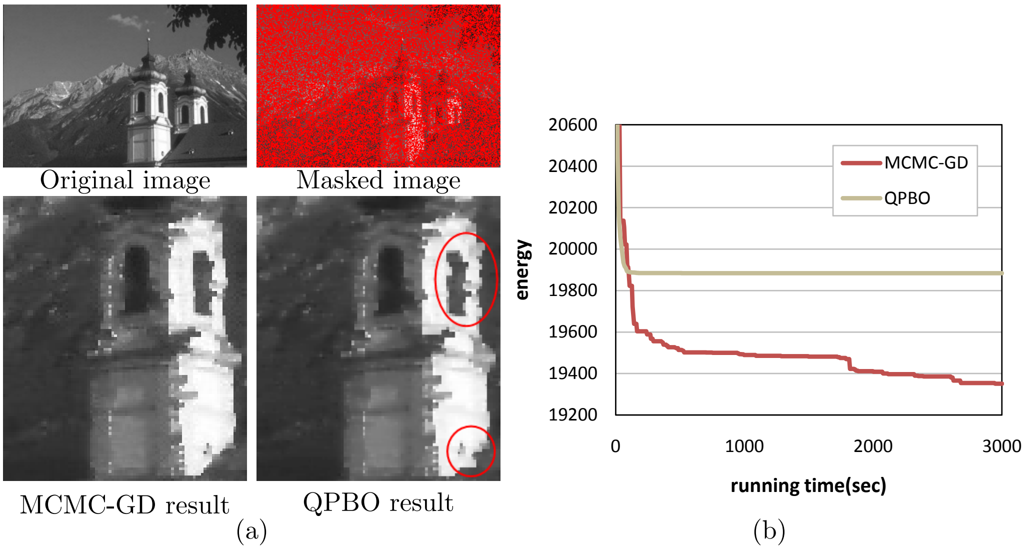


Figure 3.17: The inpainting results of the Berkeley003 image. (a) Qualitative and (b) quantitative comparisons of QPBO and MCMC-GD.

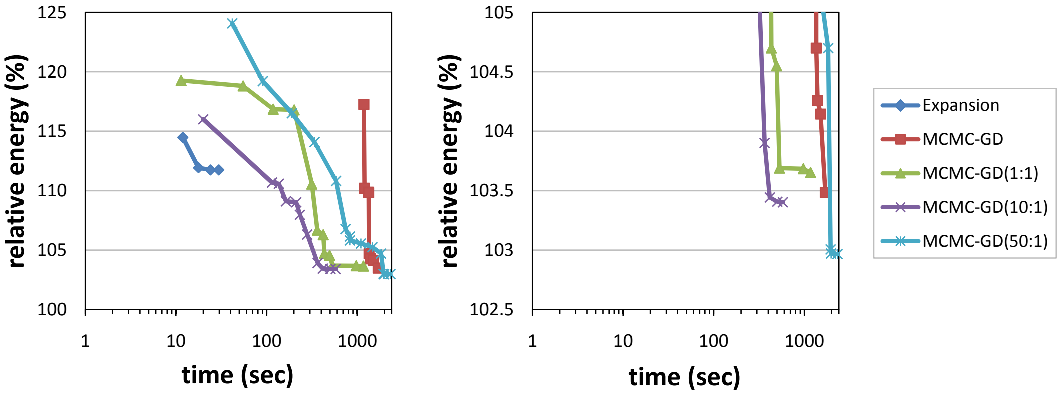


Figure 3.18: The comparison between the original MCMC-GD and its variations on Lab images. Because of the running time for calculating static anchors, the sampling procedure starts after 1000 s in the original MCMC-GD. In the alternative approaches, static anchor generation and MCMC sampling is performed in parallel. In MCMC-GD(M:N), M iterations of the message passing and N iterations of the sampling is alternatively performed. This alternative approach shortens the total running time of MCMC-GD and obtains similar results. Note that two graphs shows the same result in different scale.

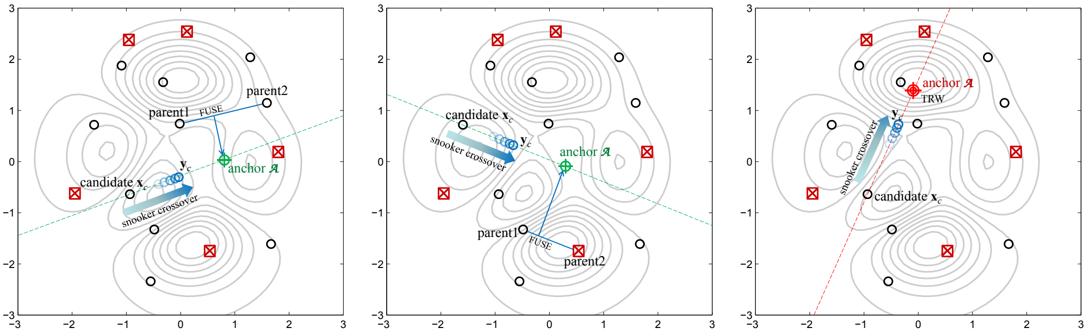
Chapter 4

Fusion Move Driven Markov Chain Monte Carlo

4.1 Introduction

MRF model have achieved great success in many vision applications [2]. Although most of them have been formulated as discrete labeling problems, continuous formulation of the problem often achieves great improvement on the qualities of the solutions in some applications such as stereo matching and optical flow.

However, continuous formulation make it much more difficult to optimize the target function compared to the fact that we have many powerful discrete optimizer such as Graph Cuts [12, 14, 13] and Message Passing methods [18, 17]. There are two dominating approaches to solve the continuous optimization problems. First approach is to model the problems as convex [52] which are easy to optimize. Despite of its success, it is limited by the fact that it cannot allow non-convex energy models. Second approach is to apply powerful discrete optimization algorithms to some re-



(a)

(b)

(c)

Figure 4.1: Three types of anchor generations and snooker crossover: (a) dynamic anchor with both parents from the samples, (b) dynamic anchor with one parent from the samples and the other from the proposals, and (c) static anchor.

duced discrete solution spaces [5, 53]. This approach is also limited by the fact that it cannot fully explore the original solution space. So it only provides approximated solutions and cannot sufficiently lower the energy of the solutions.

In this chapter, we propose a powerful optimization technique that directly solves the continuous MRF problems. It combines two powerful methods: Markov Chain Monte Carlo method (MCMC) and Quadratic Pseudo-Boolean Optimization (QPBO) fusion move. The idea to combine stochastic methods and deterministic methods was proposed by Kim and Lee [29, 31]. They, however, applied their method only to the discrete MRF optimization. To deal with continuous MRF optimization, we propose a new method, called *Fusion Move driven Markov Chain Monte Carlo* (MCMC-F). It exploits powerful deterministic methods in the framework of sampling-based stochastic method. The samples can rapidly move to lower energy state owing to deterministic methods. Also, it can effectively jump from one basin to another over the energy barrier. Consequently, this property increases mixing rate and yields faster convergence and better solutions. Moreover, exploration is not restricted in the reduced space. To demonstrate the effectiveness of the algorithm, MCMC-F is applied to stereo matching problems.

4.2 Proposed algorithm

In this section, we briefly review the sampling-based optimization, and then we explain the detail of our MCMC-F algorithm.

4.2.1 Sampling-based optimization

Sampling-based optimization exploits the sampling method to obtain optimum solution of the energy function. MCMC methods have been used to sample from a given distribution $p(\mathbf{x})$. For the optimization problem, MCMC is often embedded into the Simulated Annealing framework. However, since most MCMC methods allow only local moves, in a large solution space it takes a very long time to reach the global optimum.

To overcome the limitations of MCMC, Population-Based MCMC (Pop-MCMC) has been applied to the vision problem [27, 28]. In Pop-MCMC, multiple samples are drawn from multiple Markov chains. To obtain the optimum sample, Parallel Tempering is used instead of Simulated Annealing. It generates multiple chains in parallel according to several different temperatures, and exchanges information among them to accelerate the mixing rate. The target distribution of ith chain is defined as follows.

$$p_i(\mathbf{x}_i) = p(\mathbf{x}_i)^{\frac{1}{T_i}},$$

where $\mathbf{x}_i$ is the sample of the ith chain, $p(\mathbf{x})$ is an original target distribution, and T_i is the temperature of the ith chain.

4.2.2 MCMC combined with fusion move

The main idea of MCMC-F is to combine MCMC method and the QPBO fusion move. To combine them, we follow the strategy which has been introduced by Kim and Lee [29, 31]. Using this strategy, we have designed the algorithm to optimize the energy function formulated in section II.

As in Pop-MCMC, we derive multiple samples from multiple chains. Each sample represents a single disparity map. Those samples are iteratively updated to obtain the optimum solution in the main body of the MCMC-F algorithm. The main body is composed of two phases: anchor generation and snooker crossover. Overall algorithm of MCMC-F is summarized in Algorithm 3.

Initialization We initialize samples using plane fitting method for stereo matching. We firstly obtain 14 segmentation results of the reference (left) image using different methods [43, 54] and different parameters. And then we calculate initial disparities for each pixel using the Sum of Absolute Difference (SAD) and Winner

Algorithm 3 MCMC-F algorithm

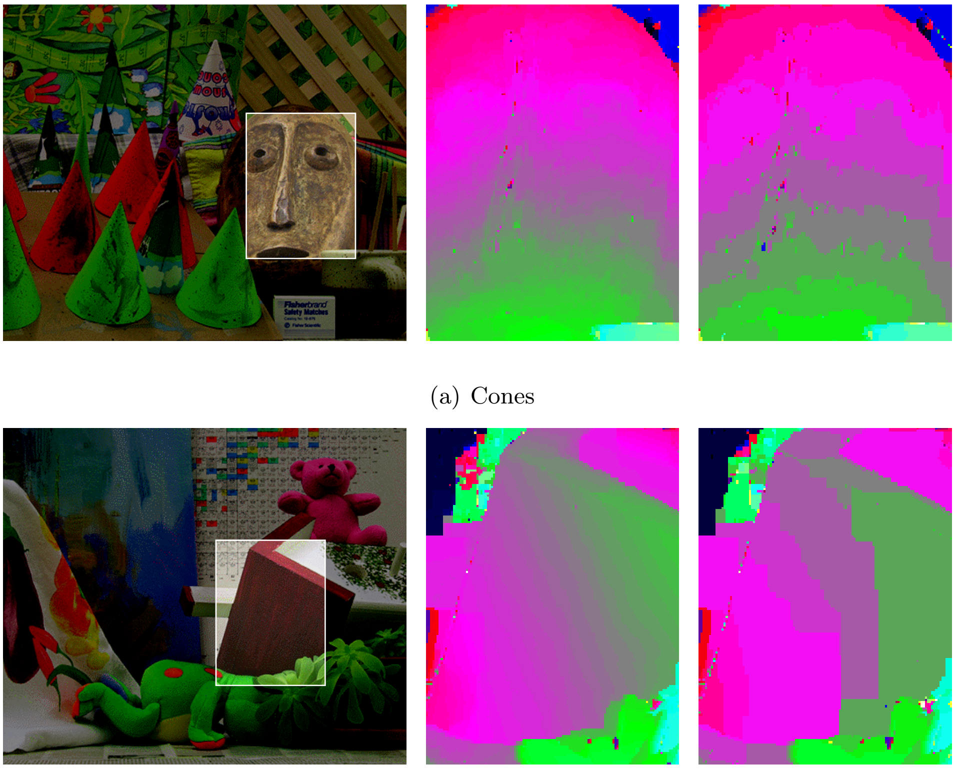
```
Initialize the population $\mathbf{x}_1, \mathbf{x}_2, \cdots$ using the proposals
repeat
    Generate an anchor $\mathcal{A}$
    Perform snooker crossover
until The algorithm converges.
```

Takes All (WTA) schemes. Incorrectly estimated disparities are eliminated by the disparity crosschecking method [41]. After that, for each segment we estimate a plane by the least squares method with reliable disparity values. After we estimated the plane, we further identify more outlier disparities that do not fit to the plane. We iteratively re-estimate the exact plane only using remaining inliers. We finally obtain 14 different disparity maps which are called proposals henceforth. Each sample in MCMC-F is initialized to one of the proposal disparity maps at random.

Anchor generation In the MCMC-F algorithm, there are three different types of anchors. The first two are the dynamic anchors. When generating dynamic anchors, we first select two samples as parents. And then, we generate a new disparity map through QPBO fusion of the parents. The solution of QPBO is the dynamic anchor. This anchor is dynamically generated and destroyed while the algorithm is running. Instead of selecting two parents from samples, we can select one of the parent from samples and another from proposals. The third type of the anchor is the static anchor. To generate the static anchors, we initially run non-move-making deterministic algorithms such as the Tree ReWeighted Message Passing (TRW) and Belief Propagation (BP) before MCMC-F starts. Those solutions are used as the anchor points while the algorithm is running. Three types of anchor generation are illustrated in Figure 4.1.

For each iteration, we generate a dynamic anchor with the probability Q_{DM}, and a static anchor with the probability $1 - Q_{DM}$.

Snooker crossover After an anchor is chosen, we update the samples by using snooker crossover. For every samples (except parent) as a candidate $\mathbf{x}_c$, we perform



(a) Cones

(b) Teddy

Figure 4.2: Close view of the resulting disparity maps. First column is reference image, second column is results of MCMC-F, and third column is results of TRW. (Best viewed in color.)

snooker crossover with the anchor point $\mathcal{A}$. Newly generated sample $\mathbf{y}_c$ lies on the line going through the candidate and the anchor according to:

$$\mathbf{y}_c = \kappa s^r \mathbf{x}_c + (1 - \kappa s^r)\mathcal{A},$$

where κ, r and s are random variables which control the dynamics of snooker crossover. They are designed to satisfy the reversibility condition of Markov chain. In our implementation, the parameters κ and r are randomly chosen among $+1$ and

-1 with equal probability. With the probability of 0.5, the parameter s has the value $s = s_{min}$ and with the probability of 0.5, the parameter s is uniformly drawn from $(s_{min}, s_{max}]$.

After the new sample $\mathbf{y}_c$ is generated, the candidate $\mathbf{x}_c$ is substituted with $\mathbf{y}_c$ according to the acceptance probability:

$$\alpha = \min\left(1, \frac{p_c(\mathbf{y}_c|I_0, I_1)}{p_c(\mathbf{x}_c|I_0, I_1)}\right).$$

where p_c is the target distribution of cth chain.

4.3 Experiments

To evaluate the performance of MCMC-F on stereo problem, we design a posterior probability of the disparity map that is composed of two terms: data term and smoothness term. The formulation is as follows.

$$p(\mathbf{x}|I_0, I_1) = \frac{1}{Z}\prod_{p\in\mathcal{V}} \exp\{-d_p(x_p)\} \cdot \prod_{(p,q)\in\mathcal{E}} \exp\{-V_{pq}(x_p, x_q)\}, \tag{4.1}$$

where $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ is the disparity map, x_p denotes the disparity of the pixel p, I_0 and I_1 is left and right images, and the set $\mathcal{V}$ and $\mathcal{E}$ contain the nodes and the edges in the MRF model, respectively.

The data term d_p is defined as follows.

$$d_p(x_p) = \|I_0(p) - I_1(p + x_p)\|^2,$$

where $\|\cdot\|$ is the Euclidean distance of the RGB color values. Since the disparity x_p can have real value, we linearly interpolate the pixel values to compute $I_1(p + x_p)$.

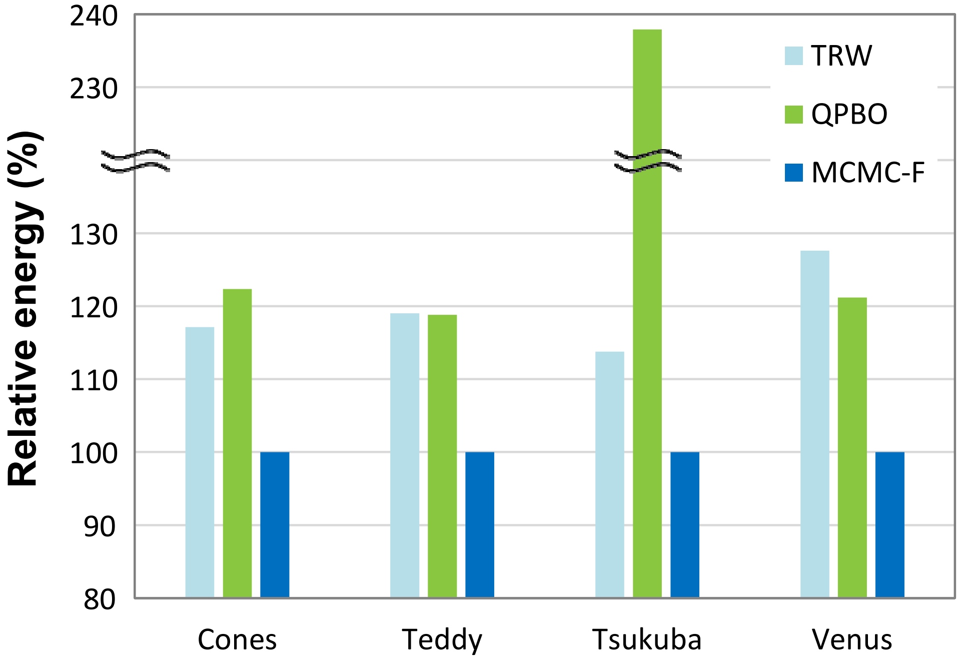


Figure 4.3: Final energy obtained by each algorithm.

For the smoothness term V_{pq}, we assume 4-neighborhood system, and we define

$$V_{pq}(x_p, x_q) = \lambda_{pq} \log \left(1 + \frac{1}{2\nu^2} |x_p - x_q|^2 \right).$$

We design the smoothness penalty to be the negative log of a Student-t function. The parameter ν controls the degree of freedom. The weight λ_{pq} varies according to the color difference between neighboring pixels so that we enforce disparity discontinuities to coincide with image color discontinuities.

To demonstrate the performance of the proposed method, we experimented on 4 Middlebury test images: Cones, Teddy, Tsukuba, and Venus [1]. We compared the performance of MCMC-F, QPBO, and TRW [18, 17]. For TRW, the original problems were reduced so that it had only integer values as labels since we cannot directly apply TRW to the continuous optimization problems. The solution of TRW was used for the static anchor. The running time of MCMC-F, and TRW was set to 1,000 seconds. QPBO took 75 $\sim$ 90 seconds to terminate. All the experiments were performed on the Intel Quad Core 2.4GHz PC platform. In all experiments, the parameters was fixed as follows: for MCMC, population size was 100, Q_{DM} was set to 0.9, the temperature of ith chain was set to $i \times 100$; for the energy function, $\nu = 0.2$ and $\lambda_{pq} = 150$ if the sum of absolute differences between $I_0(p)$ and $I_0(q)$ was less than or equal to 30, and $\lambda_{pq} = 50$ otherwise.

Some of the resulting disparity maps are shown in Figure 4.2. Final energies and the error rates of the disparity maps are depicted in Figure 4.3 and Figure 4.4, respectively. Relative energies were computed by $100 \times (\textit{energy of solution})/(\textit{minimum energy})$. We note that the results of TRW have severe artifacts due to quantization. Because the solution of TRW can have only integer values as labels, disparities of the slanted objects are presented as piece-wise fronto parallel planes. TRW cannot fully explore the continuous solution spaces. On the contrary, MCMC-F produces much more natural disparity maps. It is also shown that MCMC-F always finds the lowest energy solutions which have minimum error rates.

The properties of the solution spaces differ from image to image. For example, TRW solutions have lower energy values than QPBO solutions for Cones and Tsukuba images. On contrary, TRW solutions have higher energy values for Teddy and Venus images. Despite of that, MCMC-F achieves the lowest energy among all the algorithms for every test image.

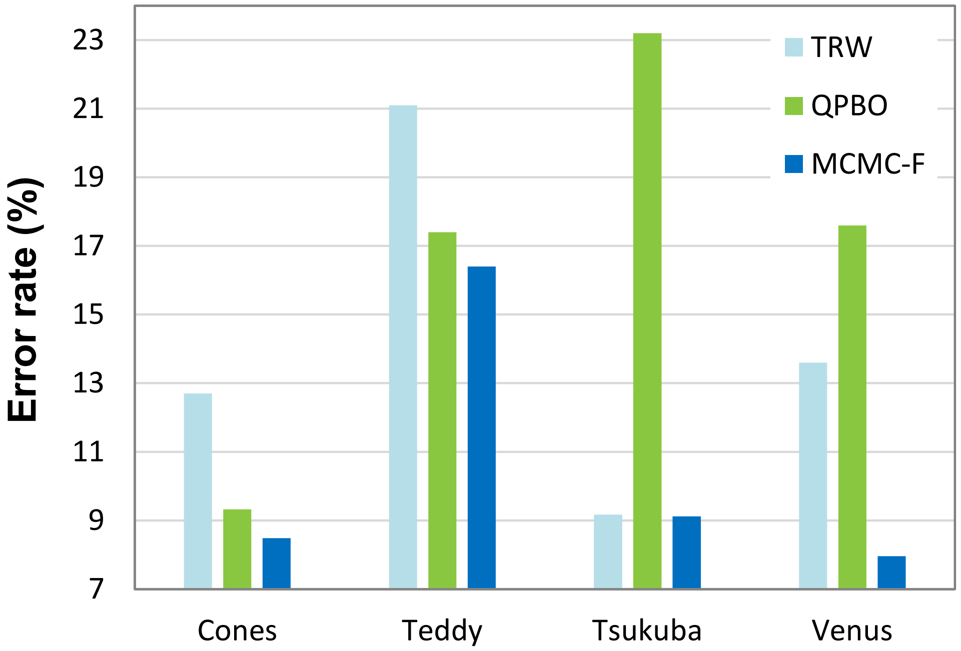


Figure 4.4: Error rate on the un-occluded region of each disparity map with error threshold of 0.5.

4.4 Summary

MRF optimization has achieved great success over wide range of applications. In some applications such as stereo matching and optical flow, continuous formulation enables to obtain much more realistic results. However, so far, existing continuous MRF optimization methods have limitations in application to real problems. In this chapter, we proposed MCMC-F algorithm to effectively optimize functions in continuous spaces. It is sampling-based optimization algorithm that is combined with fusion move. It exploits powerful discrete optimization methods while it has ability to fully explore the continuous solution space. Since samples can move to lower energy state where it is impossible to be reached by the conventional approaches, it can find lower energy state than other algorithms. We experimentally demonstrate that it achieves significantly lower energy state solutions than QPBO and TRW. Furthermore, since MCMC-F can be applied to any type of general energy functions, it will be useful to solve many other applications. And this work will also help to design more realistic complex energy models.

Chapter 5

Fusion with Graph Approximation

5.1 Introduction

Markov random field (MRF) has been used for numerous areas in computer vision [2]. MRFs are generally formulated as follows. Given a graph $G = (\mathcal{V}, \mathcal{E})$, the energy function of the pairwise MRF is given by

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} \theta_p(x_p) + \lambda \sum_{(p,q) \in \mathcal{E}} \theta_{pq}(x_p, x_q), \tag{5.1}$$

where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, and $x_p \in \{1, 2, \cdots, L\}$ is the label assigned on node p. Optimization of the MRF model is challenging because finding the global minimum of the energy function (5.1) is NP-hard in general cases.

Graph cuts-based algorithms have attracted much attention as an optimization method for MRFs [14, 13, 12, 55, 56]. Graph cuts can obtain the exact solution in polynomial time when the energy function (5.1) is submodular. Even if the function

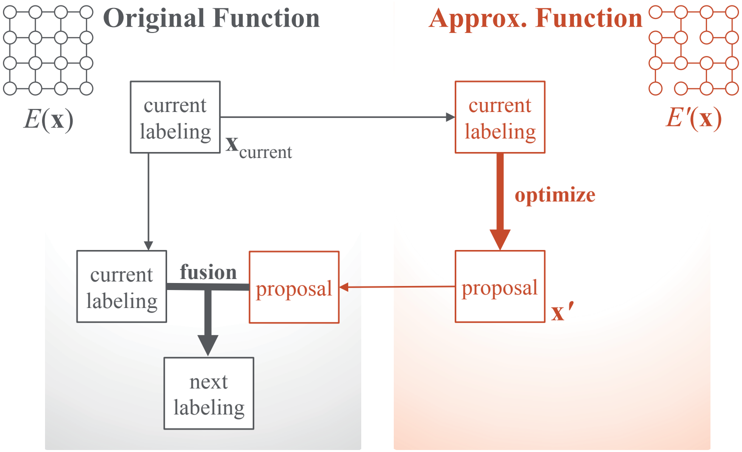


Figure 5.1: The basic idea of the overall algorithm. The original function is approximated via graph approximation. The approximated function is optimized, and the solution is used as a proposal for the original problem.

is not submodular, a partial solution can be obtained with unlabeled nodes using quadratic pseudo-Boolean optimization (QPBO) [57, 15]. Graph cuts have also been used to solve multi-label energy functions. For this purpose, move-making algorithms have been proposed, in which graph cuts optimize a sequence of binary functions to make moves.

In a move-making algorithm, the most important decision is the choice of appropriate move-spaces. For example, in α-expansion[1], move-spaces are determined by the selected α value. Simple α-expansion strategy has obtained satisfactory results when the energy function is metric. Recently, α-expansion has been shown to improve when the proper order of move-space α is selected instead of iterating a pre-specified order [58].

However, α-expansion does not work well when the energy function is non-metric.

[1] In this chapter, α-expansion always refers to QPBO-based α-expansion unless noted otherwise.

In such a case, reduced binary problems are no longer submodular. Performance is severely degraded when QPBO leaves a considerable number of unlabeled nodes. To solve this challenge, we need more elaborate proposals rather than considering homogeneous proposals as in α-expansion. Fusion move [59] can be applied to consider general proposals. In this case, we have much more choices for move-spaces.

Generating appropriate proposals is necessary for the success of fusion algorithm. However, although there has been a demand for a generic method of proposal generation [59], little research has been done on the mechanism of "good" proposal generation (we will specify the notion of "good" proposals in the next section). Instead, most research on proposal generation is often limited to heuristic and application-specific approaches [5, 9].

In this chapter, we propose a generic and application-independent approach to generate "good" proposals for non-submodular energy functions. With these proposals, we present a graph cuts-based move-making algorithm called GA-fusion (fusion with graph approximation-based proposals). This method is simple but powerful. It is applicable to any type of energy functions. The basic idea of our algorithm is presented in Figure 5.1. Section 5.3 describes the algorithm in detail.

We test our approach in real and synthetic problems. Section 5.4 demonstrates that the proposed approach outperforms other methods, particularly when the energy function is difficult. We apply our algorithm to image deconvolution and texture restoration in which conventional approaches often fail to obtain viable solutions because of strong non-submodularity. We also evaluated our algorithm on synthetic problems to show robustness to the various types of energy function.

5.2 Related works

5.2.1 Graph cuts-based move-making algorithm

Graph cuts-based algorithms have a long history. These algorithms have extended the class of applicable energy functions from binary to multi-label, from metric to non-metric, and from pairwise to higher-order energies (among these, higher-order energies are not the main concern of this chapter).

Graph cuts can obtain the global minimum when the energy function (5.1) is submodular. In binary case, a function is submodular if every pairwise term satisfies $\theta_{00} + \theta_{11} \leq \theta_{01} + \theta_{10}$, where θ_{00} represents $\theta_{pq}(0,0)$.

Graph cuts have also been successfully applied to multi-label problems. One of the most popular schemes is α-expansion. α-Expansion reduces optimization tasks into minimizing a sequence of binary energy function

$$E_b(\mathbf{y}) = E(\mathbf{x}_b(\mathbf{y})), \tag{5.2}$$

where $E_b(\mathbf{y})$ is the function of a binary vector $\mathbf{y} \in \{0,1\}^{|\mathcal{V}|}$, and $\mathbf{x}_b(\mathbf{y})$ is defined by

$$x_{b,p}(y_p) = (1 - y_p) \cdot x_p^{cur} + y_p \cdot \alpha, \tag{5.3}$$

where $x_{b,p}(y_p)$ is an element-wise operator for $\mathbf{x}_b(\mathbf{y})$ at node p, and x_p^{cur} denotes the current label assigned on node p. The label on node p switches between the current label and α according to the value of y_p. In such a case, the binary function $E_b(\mathbf{y})$ is submodular if the original function is metric [12]. This condition is relaxed in [14] such that the binary function $E_b(\mathbf{y})$ is submodular if every pairwise term satisfies

$$\theta_{\alpha,\alpha} + \theta_{\beta,\gamma} \leq \theta_{\alpha,\gamma} + \theta_{\beta,\alpha}. \tag{5.4}$$

α-Expansion is one of the most acclaimed methodologies; however, standard α-expansion is not applicable if the energy function does not satisfy condition (5.4). In such a case, a sequence of reduced binary functions is no longer submodular. We may truncate the pairwise terms [2, 50] to optimize these functions, thereby making every pairwise term submodular. This strategy works only when the non-submodular part of the energy function is very small. If the non-submodular part is not negligible, performance is seriously degraded.

For the second option, QPBO-based α-expansion can be used. In this approach, QPBO is used to optimize sub-problems of α-expansion (*i.e.*, reduced binary functions). QPBO gives optimal solutions for submodular binary functions; it is also applicable to non-submodular functions. For non-submodular functions, however, QPBO leaves a certain number of unlabeled nodes. Although QPBO-based α-expansion is usually considered as a better choice than the truncation, it also performs very poorly when the reduced binary functions have a strong non-submodularity, which creates numerous unlabeled nodes.

For the third option, QPBO-based fusion move can be considered [59]. Fusion move is a generalization of α-expansion. That produces binary functions in a way similar with α-expansion (Equation (5.2)). The only difference is the operator $x_{b,p}(y_p)$, which is defined as follows:

$$x_{b,p}(y_p) = (1 - y_p) \cdot x_p^{cur} + y_p \cdot x_p^{pro}, \tag{5.5}$$

where x_p^{pro} is a proposal labeling at node p. The value of x_p^{pro} can be different for each node contrary to the case in α-expansion. In this case, the function $E_b(\mathbf{y})$ is not always guaranteed to be submodular.

5.2.2 Proposals for fusion approach

When the fusion approach is considered, the immediate concern is related to the generation of the proposals. The choice of proposals changes move-spaces as well as the difficulties of the sub-problems, by changing the number of non-submodular terms, which consequently affects the qualities of the final solutions.

Although choosing appropriate proposals is of crucial importance, little research has been conducted on generating good proposals. In most cases, proposals are generated through heuristic and application-specific methods. For example, Woodford [5] used approximated disparity maps as proposals for stereo application. Ishikawa [9] blurred the current labeling to generate proposals for denoising application. However, these proposal generations cannot be easily applied to other applications.

Recently, Ishikawa [8] proposed an application-independent method to generate proposals. This method uses gradient descent algorithm, which can be applied to some cases, but is still limited to differentiable energy functions. Thuse, this method cannot be applied even to Potts model, which is one of the most popular prior models. In our understanding, this algorithm is only meaningful for ordered labels that represents physical quantities.

Lempitsky *et al.* pointed out two properties for "good" proposals: *quality* of individual proposal and *diversity* among different proposals [59]. In addition, we claim in this chapter that *labeling rate* is another important factor in measuring the quality of a proposal.

The three properties for good proposals are summarized in follows:

- **Quality** Good proposals are close to minimum such that proposals can guide the solution to minimum by fusion moves. In other words, good proposals have low energy.

- **Diversity** For the success of the fusion approach, diversity among different proposals is required.
- **Labeling rate** Good proposals result in high labeling rate when they are fused with the current solution. In other words, good proposals produce easy-to-solve sub-problems.

Note that these conditions are not always necessary. One may think of proposals that do not meet the foregoing conditions, but help obtain a good solution. However, in general, if proposals satisfy these conditions, we can expect to obtain a good solution. In Section 5.4, we empirically show that our proposal exhibits the above properties.

5.3 Proposed algorithm

5.3.1 Stochastic approximation

Stochastic approximation algorithms are a set of methods which optimize an objective function $f(\mathbf{x})$, which cannot be directly calculated, but only estimated via some approximations. A famous example of stochastic approximation is the stochastic gradient method.

Stochastic gradient method deals with an objective function that has the form of a sum: $f(\mathbf{x}) = \sum_i f_i(\mathbf{x})$.

For each iteration, this method updates the solution by the following equations.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla f_i(\mathbf{x}), \tag{5.6}$$

where η is a step size. This process is similar to that of gradient descent method except the fact that it calculates the approximated gradient only using a single instance of $f_i(\mathbf{x})$. Sometimes, a subset of $f_i(\mathbf{x})$ is used instead of a single instance. This method is often used for training parameters with large data set. The aim for training is to find parameters $\mathbf{x}$, which minimize loss function $f(\mathbf{x})$. However, calculation of the gradient $\nabla f(\mathbf{x})$ is often computationally expensive. In this case, we can consider stochastic gradient method using the update scheme (5.6). When the objective function is convex or pseudoconvex, appropriate scheduling of the step size η leads solution to the global minimum [26].

Our proposed algorithm is motivated by this strategy. In the context of stochastic gradient method, the need of approximation is to reduce computational complexity. On the other hand, for MRF optimization, we have another reason to use approximation. We approximate an objective function to alleviate the problem of non-submodularity. It is described in the following section how we approximate the energy function (5.1).

5.3.2 Graph approximation

We approximated an original objective function (5.1) to relieve difficulties from non-submodularity. Our motivation comes from the well-known fact that less connectivity of the graph makes fewer unlabeled nodes [15].

We exploit graph approximation by edge deletion to obtain an approximated function. This approximation is applicable to any class of energy functions, yet they are simple and easy. In graph approximation, a graph $G = (\mathcal{V}, \mathcal{E})$ is approximated as $G' = (\mathcal{V}, \mathcal{E}')$.

More specifically, we approximate the original graph with a random subset $\mathcal{E}'$

of edges from the original edge set $\mathcal{E}$. Pairwise terms θ_{pq}, where $(p, q) \in \mathcal{E} \backslash \mathcal{E}'$, are dropped from the energy formulation (5.1). The approximated function is given by the following.

$$E'(\mathbf{x}) = \sum_{p \in \mathcal{V}} \theta_p(x_p) + \lambda \sum_{(p,q) \in \mathcal{E}'} \theta_{pq}(x_p, x_q). \tag{5.7}$$

This approximation satisfies the foregoing two conditions. The approximated function is easier to solve than the original one. In other words, more nodes are labeled when we apply simple α-expansion algorithm. In addition, it remains similar to the original function. This claim is supported by the experiments in the next section.

There have been other approaches to approximate the original function in restricted structures. Some structure are known to be tractable, such as bounded treewidth subgraphs (*e.g.* tree and outer-planar graph) [18, 17, 60, 61]. However, our approximation is not restricted to any type of special structure.

The inappropriateness of these approximations to our framework can be attributed to two main reasons. First, the approximation with the restricted structures requires the deletion of too many edges. For example, tree structures have $|\mathcal{V}| - 1$ edges, and 2-bounded treewidth graphs have at most $2|\mathcal{V}| - 3$ edges. In practice, the number of edges are usually smaller than $2|\mathcal{V}| - 3$. It is not desirable scenario particularly with highly connected graphs. Second, exact optimization of 2-bounded treewidth graphs requires too much time. Several seconds to tens of seconds my be needed on the moderate size of graphs typically used in computer vision [62, 61]. Therefore, embedding this structure to our iterative framework is not appropriate.

In experimental section, we investigate the approximation with spanning trees and find it severely degrades performance.

5.3.3 Overall algorithm

The basic idea of the overall algorithm is depicted in Figure 5.1, which illustrates a single iteration of the proposed algorithm. Our algorithm first approximates original target function and then optimizes it to generate proposals.

A single iteration of algorithm is composed of two steps: proposal generation and fusion, as presented in Algorithms 4 and 5. To generate proposals, we first obtain an approximated function $E'(\mathbf{x})$ of the original $E(\mathbf{x})$ with $\rho \times 100$ percent of edges. Parameter ρ is randomly drawn from the uniform distribution $U(0, 1)$. The value of ρ changes for every expansion step. Thereafter, we perform a single iteration of α-expansion using the current labeling as the initial. Solution $\mathbf{x}'$ obtained through optimizing approximated function is used as a proposal for a fusion move. Note that, the approximated function $E'(\mathbf{x})$ is not fixed throughout the entire procedure, but it dynamically changes to give diversity to proposals.

To achieve three properties for "good proposals" mentioned in Section 5.2.2, two conditions are required for an approximated function $E'(\mathbf{x})$. First, the approximated function should be easy to solve although the original one $E(\mathbf{x})$ is difficult. Second, the approximated function should be similar to the original one. In other words, solution $\mathbf{x}'$ of the approximated function should have low energy in terms of the original function. Those characteristics are examined in next section.

5.3.4 Characteristics of approximated function

In this section, we experimentally show that the graph approximation strategy achieves the two aforementioned conditions. Through the approximation, solving

Algorithm 4 GA-fusion algorithm

```
initialize the solution $\mathbf{x}_{\text{current}}$
repeat
    <proposal generation>
    $\mathbf{x}_{\text{proposal}} \leftarrow \text{OptimizeGA}(\mathbf{x}_{\text{current}})$
    <fusion>
    $\mathbf{x}_{\text{current}} \leftarrow \text{FUSE}(\mathbf{x}_{\text{current}}, \mathbf{x}_{\text{proposal}})$
until the algorithm converges.
```

Algorithm 5 OptimizeGA($\mathbf{x}$)

```
initialize the solution with $\mathbf{x}$
for $i = 1 \to L$ do
    build a binary function $E_b$ for expansion with the label $i$
    $\rho \sim U(0, 1)$
    approximate $E_b$ by $E'_b$ using $\rho \times 100$ percent of randomly chosen edges
    $\mathbf{x} \leftarrow \arg\min_{\mathbf{x}} E'_b$
end for
return x
```

the function becomes easier, and the solution of the approximation has low energy in terms of original function.

We design the following experiments to meet the study objectives. First, we build the binary non-submodular energy functions on a 30-by-30 grid graph with 4-neighborhood structure. Unary and pairwise costs are determined as follows.

$$\theta_p(0) = 0, \theta_p(1) = k_p, \quad \text{or} \quad \theta_p(0) = k_p, \theta_p(1) = 0, \tag{5.8}$$

$$\theta_{pq}(x_p, x_q) = \begin{cases} 0 & \text{if } x_p = x_q, \\ s_{pq}\gamma_{pq} & \text{if } x_p \neq x_q, \end{cases} \tag{5.9}$$

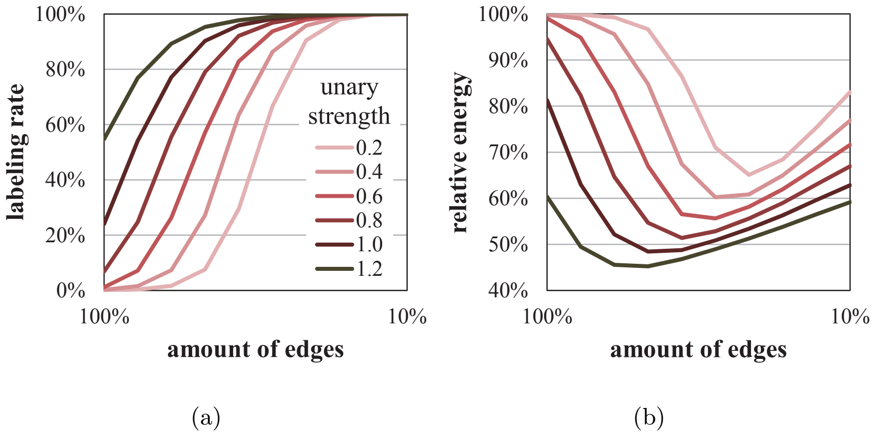


Figure 5.2: (a) Labeling rates and (b) relative energies are depicted as the graph is approximated with a random subset of edges. Relative energies are calculated with the original functions. With approximation, the labeling rate increases and the relative energy becomes lower.

where k_p and γ_{pq} are taken from a uniform distribution $U(0,1)$, and s_{pq} is randomly chosen from $\{-1,+1\}$. When s_{pq} is $+1$, the corresponding pairwise term is metric. To vary the difficulties of the problems, we control the unary strength, which is computed as $\text{mean}_{p,i}\theta_p(i)/\text{mean}_{p,q,i,j}\theta_{pq}(i,j)$ after conversion into normal form. Since above energy function is already written in normal form, it is easy to set the desired unary strength by changing the weight factor λ. The unary strength is changed from 0.2 to 1.2, with interval of 0.2. For each unary strength, 100 random instances of energy function were generated. As unary strength decreases, QPBO produces more unlabeled nodes. Of all nodes, 54.7% are labeled with the unary strength of 1.2, and none are labeled with the unary strength of 0.2.

We approximate the foregoing functions by graph approximation and then optimize them using QPBO. For approximated functions, more nodes are labeled than the original ones. The obtained solutions have low energies in terms of original func-

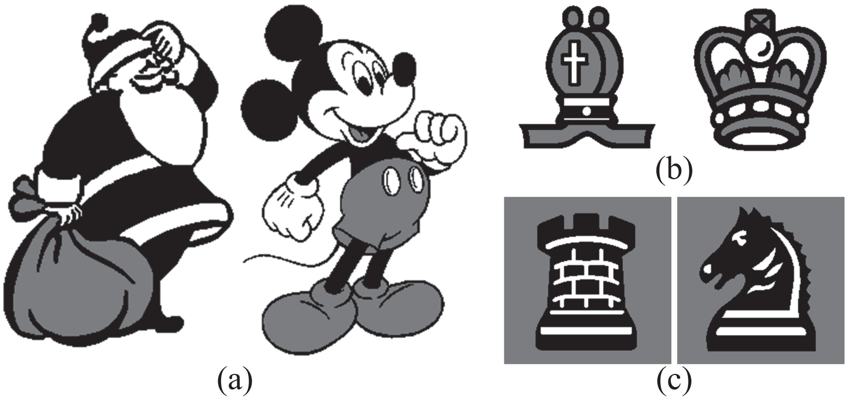


Figure 5.3: Example input images of deconvolutioin from (a) 'characters', (b) 'white chessmen', and (c) 'black chessmen' datasets.

tions. These results are summarized in Figure 5.2[2]. When the approximation uses a smaller subset $\mathcal{E}'$, more nodes are labeled. Depending on the unary strength, an appropriate size of the subset $\mathcal{E}'$ gives the lowest energy solution. Those results demonstrate that the proposed approximation makes the problem not only easy to solve but also similar to the original function.

5.4 Experiments

5.4.1 Image deconvolution

Image deconvolution is the recovery of an image from a blurry and noisy image [63]. Given its high connectivity and strong non-submodularity, this problem has been reported as a challenging one [15]. The difficult nature of the problem particularly degrades the performance of graph cuts-based algorithms. In the benchmark [15], graph cuts-based algorithms have achieved the poorest results. However, we demon-

[2] Here, relative energy is given by the energy of the solution divided by the energy of the labeling with zero for all nodes. The unlabeled nodes in the solution are labeled with zero.

strate in the following that graph cuts-based algorithm can be severely improved by the proper choice of proposals.

For experiments, we construct the same MRF model used in [63]. First, the original image (colored with three labels) is blurred with 3×3 Gaussian kernel where $\sigma = 3$. The image is again distorted with Gaussian pixel-wise noise with $\sigma = 10$. For reconstruction, the MRF model with 5×5 neighborhood window is constructed. Smoothness is given by the Potts model.

We tested various algorithms on three datasets in Figure 5.3. They include 'characters' dataset (5 images), 'white chessmen' dataset (6 images), and 'black chessmen' dataset (6 images)[3]. We compare GA-fusion with other graph cuts-based algorithms. They only differ in the strategies to generate proposals: homogeneous labeling (α-expansion), random labeling (random-fusion), dynamic programming on random spanning tree (ST-fusion), and proposed one (GA-fusion). The results imply that it is important to choose proper proposals.

We also apply other algorithms including belief propagation (BP) [16, 64], and sequential tree-reweighted message passing (TRW-S) [18, 17]. For BP and TRW-S, we used source codes provided by authors.

The results are summarized in Table 5.1. GA-Fusion always achieves lowest energy solution. Figure 5.4 shows quantitative results for the Santa image. Only GA-fusion achieved a fine result. α-Expansion converged in 3.51 seconds on average, respectively. All other algorithms are iterated for 30 seconds.

We provide more detailed analysis with the Santa image in Figures 5.5–5.7. Figure 5.5 shows the energy decrease over time in two difference scale. GA-fusion gives best performance among all tested algorithms. It is worthy of notice ST-fusion

[3] Whole data set will be provided in the supplementary material

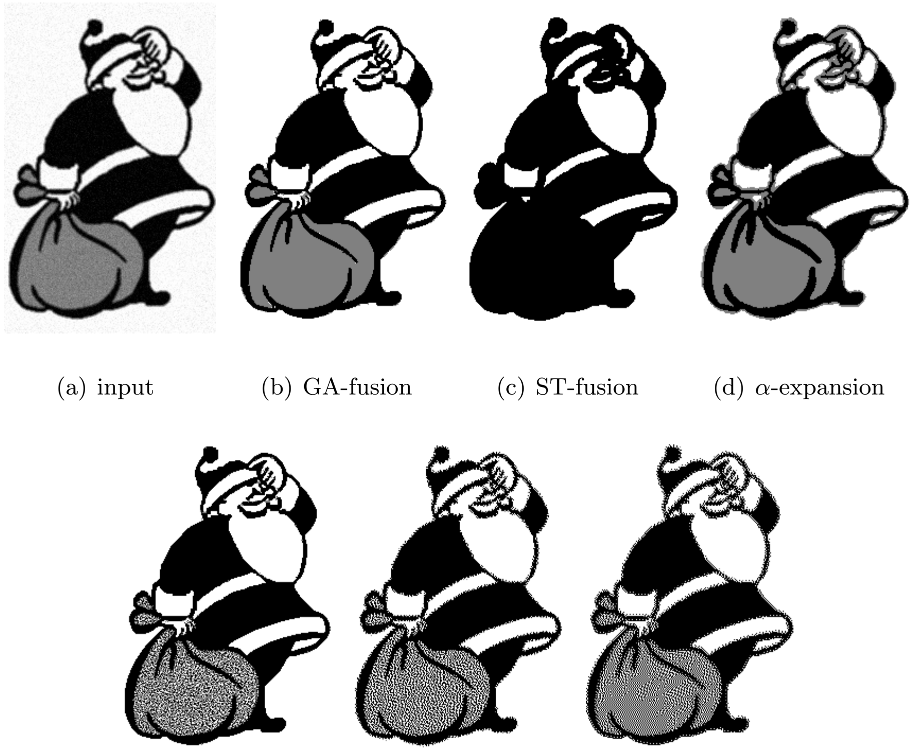
gives poor performance. Some might expect better results with ST-fusion because tree approximation makes the true optimal tractable. However, tree approximation deletes too many edges. To compare GA-proposal and ST-proposal, we generate 100 different approximated graphs of the Santa problem using our approach and another 100 using random spanning tree. We optimize former with α-expansion and latter with dynamic programming. The results are plotted on Figure 5.6. Interestingly, the plot shows a curve rather than spread. Note that tree approximation requires $\sim 92\%$ of edges to be deleted.

To figure out why our proposed method outperforms others, we provide more analysis while each graph cut-based algorithm is running (Figure 5.7). It reports the quality (energy) of the proposals and labeling ratio of each algorithm. According to section 5.2.2, "good" proposals satisfy the three conditions: quality, labeling rate, and diversity. First, GA-fusion produces the proposals with lower energy. It also achieves higher labeling rate than others. Finally, random jiggling of the plot implies that GA-fusion has very diverse proposals.

5.4.2 Binary texture restoration

The aim of binary texture restoration [65, 66] is to reconstruct the original texture image from a noisy input. Although this problem has binary labels, move-making algorithms need to be applied because QPBO often fails and gives almost unlabeled solutions.

The energy function for texture restoration is formulated as same as in [65]. Unary cost is given by $\theta_p(x_p) = -\beta/(1 + |I_p - x_p|)$, where I_p is the color of the input image at pixel p, and β is the unary cost weight. Pairwise costs are learned by computing joint histograms from the clean texture image. The costs for every edge



(a) input (b) GA-fusion (c) ST-fusion (d) α-expansion

(e) Random-fusion (f) BP (g) TRW

Figure 5.4: Image deconvolution results on the Santa image. Proposed GA-fusion algorithm achieves best results. (b–e) Four graph cuts-based algorithms obtain significantly different results. It implies that the proper choice of proposal is crucial for the success of the graph cut-based algorithm.

within window size $w = 35$ are learned first. Second, we choose a subset of edges to avoid overfitting. $S + N$ of most relevant edges are chosen, where S is the number of submodular edges, and N is the number of non-submodular edges. Relevance is given by the covariance of two nodes.

In the previous works, the numbers of edges S and N and the unary weight β were determined by learning. However, the search space of the parameters was limited because they applied conventional graph cuts and QPBO. In [65], conventional graph

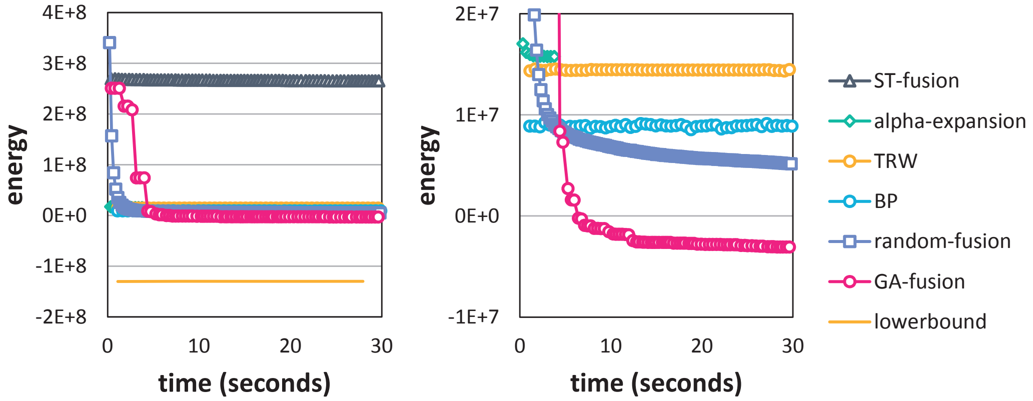


Figure 5.5: Energy decrease of each method for the deconvolution of the Santa image. Two plots shows the same curves from a single experiment, with different scales on the y-axis.

cuts are used, thus N should be fixed to zero. In [66] QPBO is used to take account of non-submodular edges. However, QPBO gives almost unlabeled solutions when N is large and β is small.

To evaluate the capability of our algorithm, we control the model parameters so that each algorithm is applied on four different settings: low-connectivity and high-unary weight; low-connectivity and low-unary weight; high-connectivity and high-unary weight; and high-connectivity and low-unary weight. For low connectivity, we use six most relevant edges ($S = 3$, $N = 3$) and for high connectivity, we use 14 most relevant edges ($S = 7$, $N = 7$). The unary weight β is chosen to be 5 and 20.

For the input, we use the Brodatz texture dataset (Figure 5.8), which contains different types of textures. Among them, 10 images are chosen for the purpose of this application. The chosen images have repeating patterns, and the size of the unit pattern is smaller than the window size (35-by-35). The images are resized to 256-by-256 pixels and binarized. Salt & pepper noise (70%) is then added.

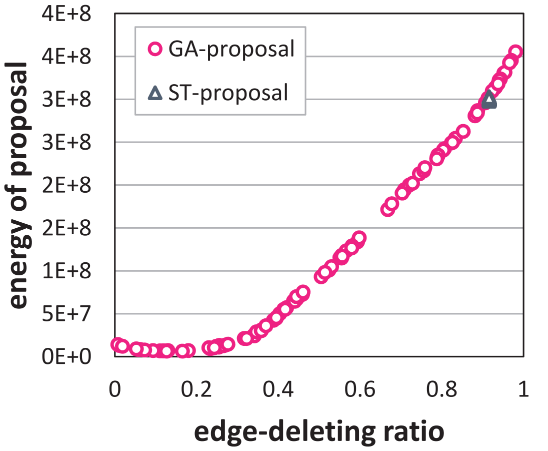


Figure 5.6: Original energy is approximated and optimized by two different methods (GA and ST). For each method 100 different random results are plotted. GA-proposals usually have lower energy than ST-proposals because random spanning tree approximation deletes too many edges.

The results are summarized in Table 5.2. Relative energies[4] are averaged over 10 texture images. When the problem is easy (low-connectivity and high-unary weight), QPBO is able to produce optimal solutions and all method except ST-fusion gives satisfactory low-energy results. Overall, GA-fusion consistently achieves low energy while others do not. QPBO and α-expansion converged in 2.28 and 3.44 seconds on average, respectively. All other algorithms are iterated for 30 seconds.

5.4.3 Analysis on synthetic problems

We compare proposed algorithm with others on various types of synthetic MRF problems to analyze performance further.

Four different types of graph structure are utilized: grid graphs with 4, 8, and 24

[4] Relative energy is calculated such that the energy of the best solution is 0 and that of zero-labeled solution is 100.

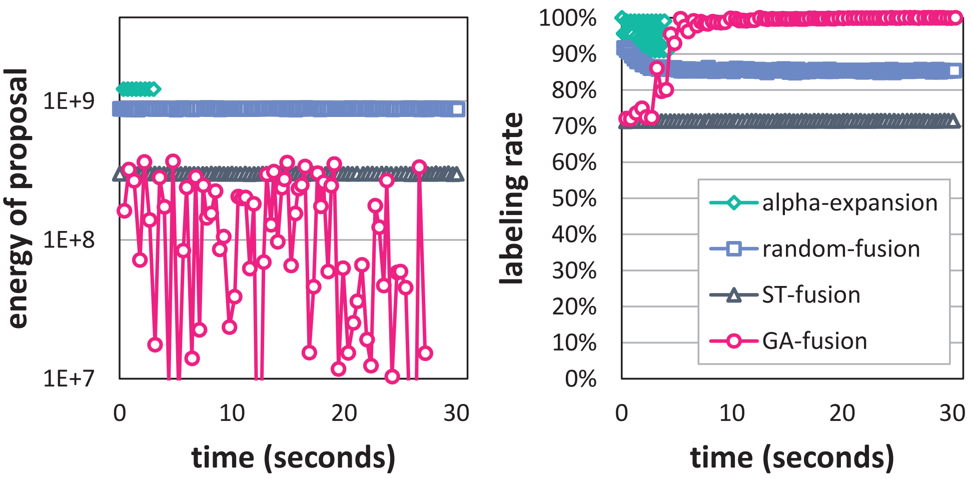


Figure 5.7: Experiment on deconvolution of the Santa image. (Left) Quality (energy) of the proposals for each iteration using a log scale. (Right) Labeling rate with the proposals for each iteration.

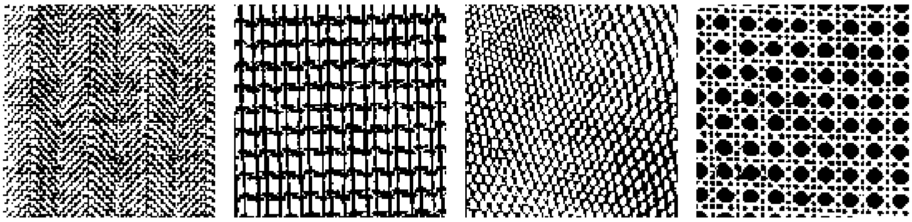
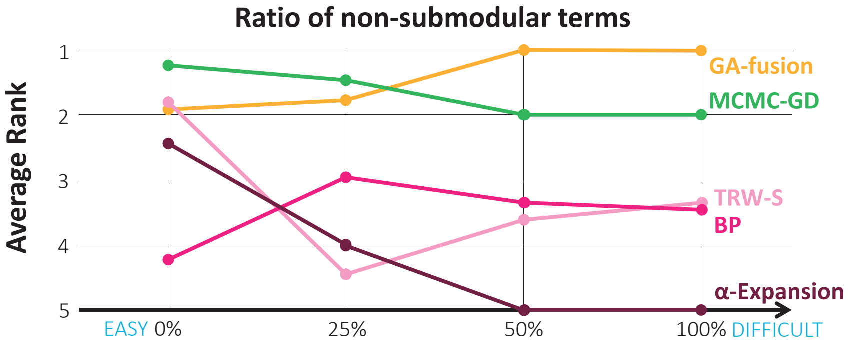


Figure 5.8: Four examples of Brodatz textures (cropped).

neighbors; and fully connected graph. The size of the grid graph is set to 30-by-30 and the size of the fully connected graph is 50. Each unary cost is assigned by random sampling from uniform distribution: $\theta_p(x_p) \sim U(0,1)$. Pairwise costs are designed using the same method in section 5.3.4 (Equation (5.9)). The difficulties of each problem are controlled by changing coupling strength λ in the energy function (5.1). The amount of non-attractive terms is set to 0%, 25%, 50% and 100% (non-attractive term means pairwise cost which does not satisfy the condition (5.4))[5]. For each graph

[5] For 4-neighborhood grid graph, 100% of non-attractive terms are impossible because by simply



MCMC-GD	35	MCMC-GD	22	GA-fusion	80	GA-fusion	60
TRW-S	29	GA-fusion	20	MCMC-GD	0	MCMC-GD	0
GA-fusion	24	BP	4	TRW-S	0	TRW-S	0
α-Expansion	18	TRW-S	2	BP	0	BP	0
BP	8	α-Expansion	1	α-Expansion	0	α-Expansion	0
	out of 40		out of 40		out of 80		out of 60

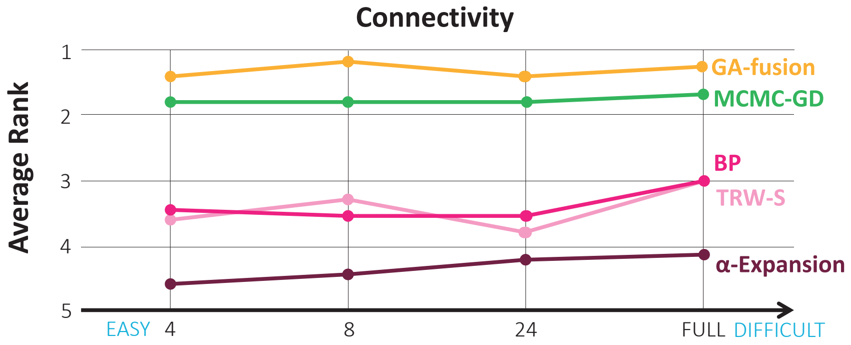
Figure 5.9: Comparison result from deferent algorithms on synthetic problems. The average ranking and the number of best case are reported according to the ratio of non-submodular terms.

setting, 10 random instances of MRF energy functions are generated. Ultimately, we construct 220 instances of MRF models.

Table 5.3 and Figure 5.9–5.11 summarize the results. Proposed two stochastic methods (MCMC-GD and GA-fusion) are compared with deterministic ones (TRW-S, BP, and α-expansion). Two important properties are reported. They are average ranking of each method and the number of the case where each method finds best solution. As shown in Table 5.3 and Figure 5.9–5.11, proposed stochastic methods outperforms the deterministic methods in overall. To figure out the performance of each algorithm according to the problem type, Figure 5.9–5.11 summarize results in different manner: according to the ratio of non-submodular terms, the connectivity, and the coupling weight.

The relative performance of each algorithm is different according to the difficul-

flipping labels every term meets the condition (5.4)

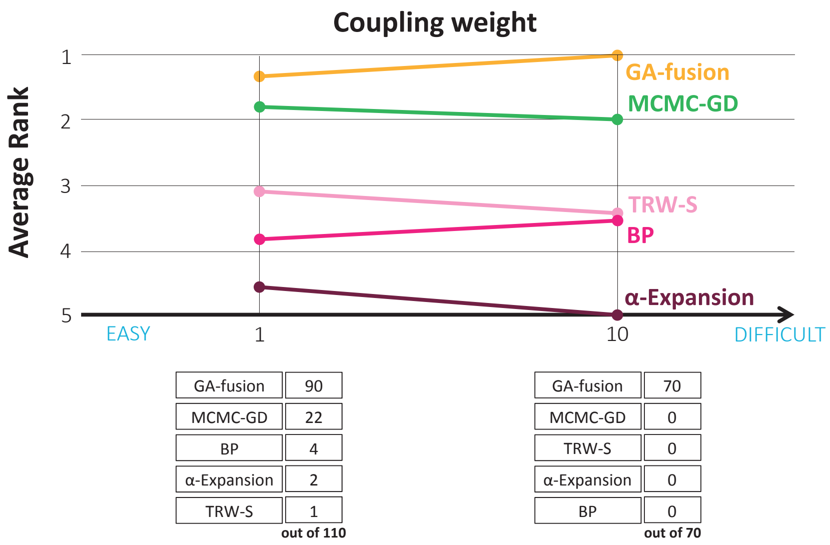


GA-fusion	33	GA-fusion	53	GA-fusion	46	GA-fusion	52
MCMC-GD	10	MCMC-GD	15	MCMC-GD	13	MCMC-GD	19
TRW-S	8	TRW-S	9	TRW-S	5	BP	11
α-Expansion	2	α-Expansion	6	α-Expansion	3	α-Expansion	9
BP	0	BP	0	BP	1	TRW-S	8
	out of 40		out of 60		out of 60		out of 60

Figure 5.10: Comparison result from deferent algorithms on synthetic problems. The average ranking and the number of best case are reported according to the connectivity.

ties of the problems. The difficulties are controlled by changing the ratio of non-submodular terms, connectivity, and coupling weight. Although GA-fusion achieves best performance in terms of average rank, it is different when it comes to submodular energy functions. As shown in Figure 5.9, the ranking of GA decrease slightly on the problem set of which the ratio of non-submodular terms are 0% and 25%. In those cases, MCMC-GD achieves best results. When difficulties are controlled by connectivity and coupling weight, the relative performance tends to be similar across different settings.

The followings are some details on the experimental settings. Graph cut-based algorithms start from the zero-labeled initial. Every algorithm, except α-expansion, is run for 10 sec because they do not follow a fixed rule for convergence. The experiment shows that 10 sec is enough time for every algorithm to converge. Although



GA-fusion	90
MCMC-GD	22
BP	4
α-Expansion	2
TRW-S	1

out of 110

GA-fusion	70
MCMC-GD	0
TRW-S	0
α-Expansion	0
BP	0

out of 70

Figure 5.11: Comparison result from deferent algorithms on synthetic problems. The average ranking and the number of best case are reported according to the coupling weight.

α-Expansion is fast, converging in less than a second, it mostly ended up with an zero-label. It is because that reduced sub-problem is too difficult and QPBO produces none of the labeled nodes in most cases.

To further investigate the superiority of proposed methods, more comparison experiments are performed. Since deterministic methods are highly dependent on the initialization, α-expansion is applied on the same set of problems while changing initialization. We tried random initialization and winner-takes-all initialization. Although final energies of α-expansion changes according to the initialization, there was no significant changes on average ranking.

We also examine how stable to the initialization the algorithm is to. To this end, GA-fusion, MCMC-GD, and α-expansion is applied to the single instance of the problem 100 times with different random initialization. Standard deviation is measured to check stability. First instances was chosen to have 50% of non-submodular terms, coupling weight 1, and connectivity 8. The standard deviation of final energies was 1.7 for GA-fusion, 0.8 for MCMC-GD, and 17.2 for α-expansion. Second instances was chosen to have 50% of non-submodular terms, coupling weight 10, and connectivity 24. The standard deviation of final energies was 101.1 for GA-fusion, 2.2 for MCMC-GD, and 253.5 for α-expansion. In both cases, proposed methods have lower standard deviation than α-expansion. MCMC-GD has lower standard deviation than GA-fusion. On second case, this difference was more significant because it is more difficult case than the first. We can conclude that the stochastic methods are more stable to initialization and sampling-based method has stronger stability.

5.5 Summary

Graph cuts-based algorithm is one of the most acclaimed algorithms for optimizing MRF energy functions. They can obtain the optimal solution for a submodular binary function and give a good approximation for multi-label function through the move-making approach. In the move-making approach, appropriate choice of the move space is crucial to performance. In other words, good proposal generation is required. However, efficient and generic proposals have not been available. Most works have relied on heuristic and application-specific ways. Thus, this chapter proposed a simple and application-independent way to generate proposals. With this proposal generation, we present a graph cuts-based move-making algorithm called GA-fusion, where the proposal is generated from approximated functions via graph approximation. We tested our algorithm on real and synthetic problems. Our experimental results show that our algorithm outperforms other methods, particularly when the problems are difficult.

Table 5.1: Image deconvolution results on four input images. Energies and average error rates are reported. The lowest energy for each case is in bold; GA-fusion achieves lowest energy for every image.

	Energy ($\times 10^6$)					
	GA	ST	α-Exp	Rand	BP	TRW-S
[characters dataset]						
Santa	**-3.06**	265.21	15.78	5.15	8.84	14.40
Pororo	**-2.70**	256.01	35.24	4.73	8.45	15.72
Mickey	**-1.27**	341.60	35.29	9.51	12.51	18.26
Rodin	**-4.14**	237.49	27.72	0.75	4.04	9.67
Gangnam	**1.86**	629.92	20.23	42.56	34.84	37.97
[white chessmen dataset]						
white king	**0.26**	197.56	16.94	3.98	6.80	10.26
white queen	**0.12**	241.12	16.11	7.47	9.40	12.48
white rook	**-0.45**	208.13	17.18	6.08	7.16	9.20
white bishop	**-0.33**	188.22	12.96	4.88	6.83	9.24
white knight	**-0.42**	213.05	13.61	7.18	8.09	9.99
white pawn	**-0.85**	126.31	8.45	3.20	4.10	5.39
[black chessmen dataset]						
black king	**2.56**	437.45	14.12	23.99	23.53	27.88
black queen	**1.96**	455.81	14.13	27.53	24.19	26.88
black rook	**1.89**	465.69	13.66	27.38	25.78	28.15
black bishop	**1.82**	496.80	13.60	31.46	27.21	29.66
black knight	**1.83**	443.54	12.78	26.96	24.16	26.56
black pawn	**1.15**	509.46	7.25	34.50	28.89	30.20
Average Error	**1.52** %	38.30%	8.18%	20.44%	34.47%	35.98%

Table 5.2: Texture restoration experiments on 10 Brodatz textures. Average of relative energies is reported. Four different types of energy are considered by changing the number of pairwise costs and unary weight. The lowest energy for each case is in bold.

Energy type	QPBO	GA-fusion	ST-fusion	α-expansion	random-fusion	BP	TRW-S
low-connectivity & high-unary weight	**0.0**	**0.0**	1.9	0.1	0.1	**0.0**	**0.0**
low-connectivity & low-unary weight	n/a	**1.5**	25.6	2.8	5.1	3.6	10.6
high-connectivity & high-unary weight	n/a	0.9	25.3	**0.1**	0.9	0.4	3.3
high-connectivity & low-unary weight	n/a	**2.2**	38.3	8.3	4.6	6.0	11.6

Table 5.3: Comparison result from deferent algorithms on synthetic problems. Average ranking for 220 problem instances is reported. Also, number of problem instances where each algorithm achieves best result is reported

	Average ranking	number of best case
GA-fusion	1.3	184
MCMC-GD	1.8	57
TRW-S	3.4	30
BP	3.4	20
α-Expansion	4.3	12

Chapter 6

Conclusion

6.1 Summary and contribution of the dissertation

MRF optimization is one of the fundamental problems in computer vision. To solve this problem, many methods have been proposed. However, because of its NR-hardness, existing methods fails when it comes to more difficult MRF optimization problems. To tackle this, four novel approaches based on stochastic optimization method are proposed in this dissertation. By adopting the stochastic framework, proposed methods explore more over the solution space and achieve better solution unlike other deterministic optimization methods, which easily get stuck at local optima.

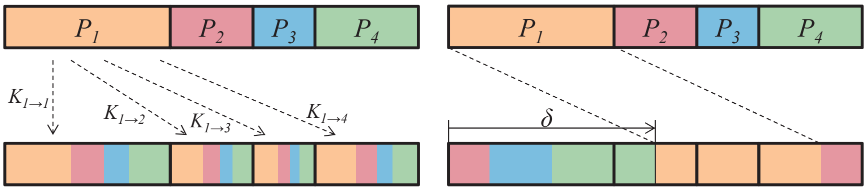
There are two different ways to embed randomness in the algorithm. The first is to allow the solution randomly move on the solution space. For example, Markov chain Monte Carlo (MCMC) methods uses probabilistic transition kernel to allow random move. The second is to randomly distort the posteriori probability function and then allow deterministic move on the approximated posteriori. For example, stochastic approximation methods approximate the original distribution in random way. In this dissertation, Pop-MCMC, MCMC-GD, and MCMC-F are proposed on the framework of MCMC. And GA-fusion is proposed on the framework of stochastic approximation.

In this dissertation, two main ideas have been proposed to develop efficient methods. The first is to adopt population-based framework. This idea is originated from the genetic algorithm and then modified to fit the MCMC framework [24]. By running multiple Markov chain and allowing them the exchange of information, the mixing rate can be improved. In consequence, MCMC converges to the better solution. Using this approach, Pop-MCMC, MCMC-GD, and MCMC-F are proposed. Second idea is to exploit the power of existing deterministic methods. It is done by embedding the deterministic methods into the framework of stochastic optimization. Using this approach, MCMC-GD, MCMC-F, and GA-fusion are developed.

6.2 Future works

6.2.1 MCMC without detailed balance

Recent studies have shown that MCMC can be accelerated by breaking the detailed balance [67, 68], which has often been considered one of the essential elements of MCMC. Although detailed balance condition provides us great simplicity in designing a kernel, it is not a necessary condition for MCMC. By breaking detailed balance, more efficient kernel is to be available. Given that detailed balance is not a necessary condition, stationary distribution is achievable even without detailed balance. However, designing MCMC kernel without detailed balance is not trivial. Recently, Suwa and Todo [67] proposed a generic framework to build a non-reversible kernel with-



(a) Gibbs sampler

(b) Suwa–Todo method

Figure 6.1: Example of landfill for the transition kernel of Gibbs sampler and Suwa–Todo method. Upper row depicts current distribution and lower row depicts the distribution after applying kernel. The transition kernel is visualized as moving boxes. Unlike Gibbs sampler, the kernel of Suwa–Todo method has the zero rejection rate in this example. (best viewed in color)

out detailed balance. Similarly, non-reversible kernel can improve the performance of MRF optimization in computer vision.

The Gibbs sampler updates a single node from its conditional distribution $p(x_i|\mathbf{x}\setminus x_i)$. Given that we are dealing with the process of updating a single node, let us omit $(x_i|\mathbf{x}\setminus x_i)$ and denote the probability as p_a when $\mathbf{x}_i$ is assigned with the label a. We denote a transition kernel as $K_{a\to b}$ where a is the current label of the node i and b is the next label. The detailed balance condition is given by the following equation.

$$p_a K_{a\to b} = p_b K_{b\to a}. \tag{6.1}$$

A transition kernel can be visually understood as a landfill model. The kernel for Gibbs sampler is depicted in Figure 6.1(a). In the landfill model, the probabilities are represented as boxes with size proportional to the probability values. These boxes move according to the transition kernel $K_{a\to b}$ while preserving the size of the probability boxes.

The Suwa–Todo method also can be easily understood by employing the landfill model. The transition kernel for the Suwa–Todo method is illustrated in Figure 6.1(b). Let us consider how the transition kernel updates a single node. First, γ is randomly chosen from the uniform distribution between C_{a-1} and C_a, where $C_a = \sum_{k=1}^{a} p_k$. The node i is updated to the label b *s.t.* $p_{b-1} < \gamma + \delta \leq p_b$ or $p_{b-1} < \gamma + \delta - 1 \leq p_b$.

6.2.2 Stochastic approximation for higher-order MRF model

Recently, there has been increasing emphasis on the higher-order MRF models because it can capture the rich statistics of natural scenes [4, 5, 8, 9, 49]. However, due to intrinsic difficulty of the model and the lack of efficient algorithms, it has often been troublesome to use the higher-order MRF models. There are some approaches to overcome those limitations. Proposed MCMC-GD algorithm was successfully applied to the higher-order model (Section 3.4.2.2). On the other hand, however, GA-fusion is only applicable to the pairwise model. To solve this problem, graph approximation method for higher-order clique should be examined.

Bibliography

[1] D. Scharstein and R. Szeliski, "Middlebury stereo vision page," http://vision.middlebury.edu/stereo/.

[2] 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, June 2008.

[3] R. S and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 860–867, June 2005.

[4] P. Kohli, L. Ladický, and P. H. Torr, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, May 2009.

[5] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second-order smoothness priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2115–2128, December 2009.

[6] S. Roth and M. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, April 2009.

[7] B. Potetz and T. Lee, "Efficient belief propagation for higher-order cliques using linear constraint nodes," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 39–54, October 2008.

[8] H. Ishikawa, "Higher-order gradient descent by fusion move graph cut," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 568–574, Sebtember 2009.

[9] ——, "Higher-order clique reduction in binary graph cut," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2993–3000, June 2009.

[10] A. M. Ali, A. A. Farag, and G. L. Gimel'Farb, "Optimizing binary MRFs with higher order cliques," in *Proceedings of European Conference on Computer Vision*, pp. 98–111, October 2008.

[11] B. Potetz, "Efficient belief propagation for vision using linear constraint nodes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[12] 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, November 2001.

[13] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124–1137, September 2004.

[14] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, February 2004.

[15] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary MRFs via extended roof duality," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[16] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 900–907, October 2003.

[17] 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, October 2006.

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

[19] N. Komodakis and N. Paragios, "Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles," in *Proceedings of European Conference on Computer Vision*, pp. 806–820, October 2008.

[20] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” in *Proceedings of European Conference on Computer Vision*, pp. 82–96, May 2002.

[21] A. Barbu and S.-C. Zhu, “Generalizing swendsen-wang to sampling arbitrary posterior probabilities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1239–1253, August 2005.

[22] A. Barbu and S. C. Zhu, “Multigrid and multi-level Swendsen-Wang cuts for hierarchic graph partition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 731–738, June 2004.

[23] S. Kirkpatrick, D. G. Jr., and M. P. Vecchi, “Optimization by simmulated annealing,” *science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[24] F. Liang and W. H. Wong, “Evolutionary Monte Carlo: Applications to c_p model sampling and change point problem,” *Statistica Sinica*, vol. 10, no. 2, pp. 317–342, April 2000.

[25] A. Jasra, D. A. Stephens, and C. C. Holmes, “On population-based simulation for static inference,” *Statistics and Computing*, vol. 17, no. 3, pp. 263–279, September 2007.

[26] L. Bottou, “Online algorithms and stochastic approximations,” in *Online Learning and Neural Networks*, D. Saad, Ed. Cambridge University Press, 1998.

[27] J. Park, W. Kim, and K. M. Lee, “Stereo matching using population-based mcmc,” in *Proceedings of IEEE International Conference on Computer Vision*, October 2007, pp. 560–569.

[28] W. Kim, J. Park, and K. M. Lee, “Stereo matching using population-based MCMC,” *International Journal of Computer Vision*, vol. 83, no. 2, pp. 195–209, June 2009.

[29] W. Kim and K. M. Lee, “Markov chain Monte Carlo combined with deterministic methods for Markov random field optimization,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1406–1413, June 2009.

[30] ——, “A hybrid approach for mrf optimization problems: Combination of stochastic sampling and deterministic algorithms,” *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1623–1637, December 2011.

[31] ——, “Continuous markov random field optimization using fusion move driven markov chain monte carlo technique,” in *Proceedings of International Conference on Pattern Recognition*, pp. 1364–1367, August 2010.

[32] ——, “Non-submodular mrf optimization by graph approximation,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, submitted*, June 2014.

[33] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, July 2003.

[34] J. Wang and R. Swendsen, “Nonuniversal critical dynamics in Monte Carlo simulations,” *Physical Review Letters*, vol. 58, pp. 86–88, January 1987.

[35] C. J. Geyer, “Markov chain monte carlo maximum likelihood,” in *Computing Science and Statistics: Proceedings of the Symposium on the Interface*, pp. 156–163, April 1992.

[36] K. Hukushima and K. Nemoto, “Exchange Monte Carlo method and application to spin glass simulations,” *Journal of the Physical Society of Japan*, vol. 65, no. 6, pp. 1604–1608, June 1996.

[37] F. Liang and W. H. Wong, “Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models,” *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 653–666, December 2001.

[38] W. M. Spears, “Crossover or mutation?” in *Proceedings of the Workshop on Foundations of Genetic Algorithms*, pp. 221–237, July 1992.

[39] H. Tao, H. S. Sawhney, and R. Kumar, “A global matching framework for stereo computation,” in *Proceedings of IEEE International Conference on Computer Vision*, pp. 532–539, July 2001.

[40] M. Bleyer and M. Gelautz, “Graph-based surface reconstruction from stereo pairs using image segmentation,” in *Proceedings of Society of Photo-Optical Instrumentation Engineers*, pp. 288–299, February 2005.

[41] L. Hong and G. Chen, “Segment-based stereo matching using graph cuts,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 74–81, June 2004.

[42] A. Klaus, M. Sormann, and K. Karner, “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure,” in *Proceedings of International Conference on Pattern Recognition*, pp. 15–18, August 2006.

[43] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[44] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “Middlebury mrf minimization,” http://vision.middlebury.edu/MRF/.

[45] M. J. A. Strens, M. Bernhardt, and N. Everett, “Markov chain Monte Carlo sampling using direct search optimization,” in *Proceedings of International Conference on Machine Learning*, pp. 602–609, July 2002.

[46] A. Barbu and S. C. Zhu, “Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1239–1253, August 2005.

[47] H. Y. Jung, K. M. Lee, and S. U. Lee, “Toward global minimum through combined local minima,” in *Proceedings of European Conference on Computer Vision*, pp. 298–311, October 2008.

[48] ——, “Window annealing over square lattice Markov random field,” in *Proceedings of European Conference on Computer Vision*, pp. 307–320, October 2008.

[49] X. Y. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black, “Efficient belief propagation with learned higher-order Markov random fields,” in *Proceedings of European Conference on Computer Vision*, pp. 269–282, May 2006.

[50] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 294–302, August 2004.

[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 416–423, July 2001.

[52] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 67, no. 2, pp. 141–158, April 2006.

[53] V. Lempitsky, S. Roth, and C. Rother, "FusionFlow: Discrete-continuous optimization for optical flow estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[54] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, September 2004.

[55] K. Alahari, P. Kohli, and P. Torr, "Reduce, reuse & recycle: Efficiently solving multi-label MRFs," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[56] P. Kohli and P. Torr, "Efficiently solving dynamic markov random fields using graph cuts," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 922–929, October 2005.

[57] P. L. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation and persistency in quadratic 0-1 optimization," *Mathematical Programming*, vol. 28, no. 2, pp. 121–155, February 1984.

[58] D. Batra and P. Kohli, "Making the right moves: Guiding alpha-expansion using local primal-dual gaps," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1865–1872, June 2011.

[59] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for Markov random field optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1392–1405, August 2010.

[60] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 384–390, June 2005.

[61] D. Batra, A. Gallagher, D. Parikh, and T. Chen, "Beyond trees: MRF inference via outer-planar decomposition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2496–2503, June 2010.

[62] A. Fix, J. Chen, E. Boros, and R. Zabih, "Approximate MRF inference using bounded treewidth subgraphs," in *Proceedings of European Conference on Computer Vision*, pp. 385–398, October 2012.

[63] A. Raj and R. Zabih, "A graph cut algorithm for generalized image deconvolution," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1048–1054, June 2005.

[64] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[65] D. Cremers and L. Grady, "Statistical priors for efficient combinatorial optimization via graph cuts," in *Proceedings of European Conference on Computer Vision*, pp. 263–274, May 2006.

[66] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts–a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1274–1279, July 2007.

[67] H. Suwa and S. Todo, "Markov chain Monte Carlo method without detailed balance," *Physical Review Letters*, vol. 105, no. 12, p. 120603, September 2010.

[68] K. S. Turitsyn, M. Chertkov, and M. Vucelja, "Irreversible monte carlo algorithms for efficient sampling," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4, pp. 410–414, February 2011.

국문초록

마르코프 랜덤 필드 모델은 컴퓨터 비전 분야에서 중요한 모델이다. 스테레오 정합, 영상 분할, 노이즈 제거, 인페인팅 등 많은 비전 문제들이 마르코프 랜덤 필드의 최적화 문제로 수식화되었다. 마르코프 랜덤 필드의 최적화 문제를 풀기 위하여 다수의 알고리즘들이 개발되어 왔다. 상대적으로 쉬운 난이도를 갖는 문제에 있어서는 많은 알고리즘이 성공적으로 적용된 반면, 어려운 문제에서는 여전히 만족스럽지 못한 성능을 보이고 있다. 그러한 어려움들은 마르코프 랜덤 필드 모델에 내재된 속성으로부터 오는데, 비서브모듈러 항, 큰 에지 계수, 고차 클릭 등이 이에 해당된다.

본 학위 논문에서는 최적화를 위한 몇가지의 알고리즘을 제시한다. 제시된 알고리즘들에 사용된 중요한 아이디어는 확률론적 방법과 결정론적 방법을 결합하는 것이다. 확률론적 방법을 이용하면 문제 공간에서의 더 넓은 탐색(exploration)이 가능하다. 반면, 결정론적 방법을 이용하면 더 효율적인 활용(exploitation)이 가능하다. 이러한 두 방법론을 결합함으로써 더욱 향상된 성능을 꾀할 수 있다. 이를 위하여 마르코프 체인 몬테 카를로 기법과 확률적 근사법을 이용한다.

먼저, 마르코프 체인 몬테 카를로 기법을 기본으로 하여 Pop-MCMC, MCMC-GD, MCMC-F로 불리는 세 가지의 알고리즘을 제안한다. 마르코프 체인 몬테 카를로 기법은 전역 최적점을 찾을 수 있는 훌륭한 이론적 배경을 마련하여 주지만, 수렴 속도가 느린 것이이 단점이다. 이를 극복하기 위하여 개체군 기반의 틀과 결정론적 방법과의 결합 전략이 이용된다. 결과적으로 기존의 기법들에 비해 더욱 빠른

수렴을 갖는 최적화 기법을 제안한다.

다음으로 이용된 확률적 근사법에서는 목적함수를 확률적으로 근사하게 되는데, 이를 마르코프 랜덤 필드 최적화에 적용하기 위해 그래프 근사를 제안한다. 그래프 근사를 통해 비서브모듈러 항으로 인한 문제를 감소시킬 수 있다. 이러한 아이디어를 기존의 그래프 컷 알고리즘과 결합하여 GA-fusion 알고리즘을 제안한다.

제안한 알고리즘의 성능을 면밀히 평가하기 위하여 다양한 실험이 시행되었다. 제안된 알고리즘은 스테레오 정합, 영상 몽타주, 인페인팅, 영상 디콘볼루션, 텍스처 복원 등 컴퓨터 비전 분야의 다양한 문제들에 적용되었다. 또한 알고리즘들의 성능을 더욱 더 자세히 분석하고자 가상의 마르코프 랜덤 필드 문제를 구성하여 문제에 난이도에 따른 특성과 알고리즘의 파라미터에 따른 특성들을 분석하였다.

주요어: 마르코프 랜덤 필드, 조합 최적화, 마르코프 체인 몬테 카를로, 개체군 알고리즘, 확률적 근사법, 비서브모듈러 에너지 모델, 고차 클릭 에너지 모델

학번: 2007-20950