การหาค่าเหมาะที่สุดของสนามสุ่มด้วยลำดับชั้นเฉพาะที่

นายสร้างสรร ลีฬหะพันธุ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Random Field Optimization using Local Label Hierarchy

Mr. Sangsan Leelhapantu

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

Thesis Title	Random Field Optimization using Local Label Hierarchy
By	Mr. Sangsan Leelhapantu
Field of Study	Computer Engineering
Thesis Advisor	Assistant Professor Thanarat Chalidabhongse, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

....Dean of the Faculty of Engineering
(Associate Professor Supot Teachavorasinskun, D.Eng.)

THESIS COMMITTEE

....Chairman
(Professor Boonserm Kijsirikul, D.Eng.)

....Thesis Advisor
(Assistant Professor Thanarat Chalidabhongse, Ph.D.)

....Examiner
(Associate Professor Chotirat Ratanamahatana, Ph.D.)

....Examiner
(Assistant Professor Sukree Sinthupinyo, Ph.D.)

....External Examiner
(Assistant Professor Kuntpong Woraratpanya, D.Eng.)

สร้างสรร ลีฬหะพันธุ์ : การหาค่าเหมาะที่สุดของสนามสุ่มด้วยลำดับชั้นเฉพาะที่ (Random Field Optimization using Local Label Hierarchy) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. ธนารัตน์ ชลิดาพงศ์, 99 หน้า.

การแปลงปัญหาให้อยู่ในรูปแบบสนามสุ่มได้รับการพิสูจน์แล้วว่าเป็นกรอบงานที่ใช้แก้ปัญหาทางด้านคอมพิวเตอร์วิทัศน์ได้ดี โดยเฉพาะอย่างยิ่งงานที่เกี่ยวกับการกำหนดป้ายกำกับที่เหมาะสมให้กับจุดภาพหรือเซ็ตของจุดภาพที่อยู่ภายใต้ความสัมพันธ์เชิงพื้นที่และบริบทเชิงการมองเห็น เนื่องด้วยความสามารถในการเชื่อมโยงสารสนเทศวงกว้างและสารสนเทศเฉพาะที่ได้อย่างเป็นธรรมชาติ อย่างไรก็ตามการแก้ปัญหาเหล่านี้อาจไม่สามารถทำได้ในทางปฏิบัติในกรณีที่มีตัวแปรสุ่มและป้ายกำกับที่เป็นไปได้จำนวนมากเนื่องจากความซับซ้อนทางการคำนวณเติบโตอย่างรวดเร็วตามขนาดของปัญหา

วิทยานิพนธ์นี้ได้เสนอวิธีการเพิ่มอัตราเร็วสำหรับการหาค่าเหมาะที่สุดของสนามสุ่มโดยใช้ลำดับชั้นเฉพาะที่ ในงานนี้ได้ให้ความสนใจกับปัญหาที่ปริภูมิป้ายกำกับมีโครงสร้างแบบอันดับซึ่งแทนปริมาณเชิงกายภาพและได้ใช้ประโยชน์จากลักษณะเฉพาะของปัญหาการติดป้ายกำกับเพื่อให้ได้มาซึ่งวิธีการหาค่าต่ำที่สุดของพลังงานแบบลำดับชั้น ทำให้สามารถหลีกเลี่ยงการค้นหาโดยแจงกรณีบนปริภูมิป้ายกำกับและได้สมรรถนะที่ดีขึ้นในเชิงเวลาการทำงาน วิทยานิพนธ์นี้ได้ให้นิยามและสัญกรณ์สำหรับลำดับชั้นเฉพาะที่และ ได้เสนอวิธีการจัดกลุ่มเชิงป้ายกำกับ ได้แก่ การค้นหาค่าต่ำที่สุดเฉพาะที่ การวิเคราะห์กลุ่ม และการแบ่งกลุ่มด้วยค่าผลต่างมากที่สุด นอกจากนี้ยังได้ขยายขอบเขตนิยามของฟังก์ชันพลังงานให้มีโดเมนครอบคลุมเซ็ตของป้ายกำกับและเสนอวิธีการในการกำหนดศักย์ของกลุ่ม ขั้นตอนการประมวลผลที่เพิ่มขึ้นมามีความซับซ้อนในการคำนวณเชิงทฤษฎีที่น้อยกว่าการประมวลผลทั้งหมดอย่างมีนัยสำคัญ

วิธีการที่นำเสนอได้ถูกประเมินผลกับปัญหาทางคอมพิวเตอร์วิทัศน์หลายปัญหาที่มีปริภูมิป้ายกำกับแบบมีโครงสร้าง ผลการทดลองได้แสดงให้เห็นว่าวิธีการที่เสนอสามารถช่วยเพิ่มอัตราเร็วการประมวลผลได้มากสุดถึงสิบเท่าโดยยังคงให้ค่าพลังงานทัดเทียมวิธีการแบบเดิม

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต ....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

ปีการศึกษา 2560

5471428421 : MAJOR COMPUTER ENGINEERING

KEYWORDS: COMPUTER VISION / RANDOM FIELD / DISCRETE ENERGY MINIMIZATION / QPBO

SANGSAN LEELHAPANTU: Random Field Optimization using Local Label Hierarchy. ADVISOR: ASST. PROF. THANARAT CHALIDABHONGSE, Ph.D., 99 pp.

Random field formulation has proven to be a powerful framework for solving various computer vision tasks, specifically those involving assigning labels to image pixels or superpixels subjected to spatial relationships and visual contexts, due to the ability to intuitively incorporate global and local information. Unfortunately, solving these problems can be impractical when large number of variables and possible labels are present as the computational complexity grows fast with the problem size.

In this thesis, we propose a speedup scheme for random field optimization using local label hierarchy. We focus on problems in which the label space has a natural ordering structure that represents physical quantity and exploit characteristics of the underlying labeling problems to obtain a hierarchical energy minimization technique. This has enabled us to circumvent exhaustive search of label space and, therefore, achieve better performance in terms of running time. We give definitions and notations for local label hierarchy as well as present approaches for label-wise grouping, namely, local minimum search, cluster analysis, and maximum-difference subdivision. We also generalize the definition of energy function to include sets of labels as the domain and present heuristics for assigning group potentials. The added processing steps have significantly less theoretical computational complexity than the overall process.

Our methodology was tested with a number of computer vision problems with structured label spaces. The experimental results have shown that our proposed scheme can provide up to an order of magnitude speedup of the computation time while providing comparable energy.

Department:	Computer Engineering	Student's Signature	....
Field of Study:	Computer Engineering	Advisor's Signature	....
Academic Year:	2017		

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisor, Asst. Prof. Thanarat Horprasert Chalidabhongse, who have given me the freedom to explore on my own while providing guidance and encouragement when I stumbled. One could not wish for a better or friendlier advisor.

I would also like to express my gratitude to the rest of my thesis committee: Prof. Boonserm Kijsirikul, Assoc. Prof. Chotirat Ratanamahatana, Asst. Prof. Sukree Sinthupinyo, and Asst. Prof. Kuntpong Woraratpanya, for their insightful comments, constructive criticisms, and hard questions.

I would also like to thank my fellow CGCI labmates, my friends and fellow students in Chulalongkorn University for friendly and cheerful daily work environment throughout the years.

The Scholarship from the Graduate School, Chulalongkorn University to commemorate the 72nd anniversary of his Majesty King Bhumibol Aduladeja is gratefully acknowledged.

Last but not least, I would also like to thank my family for the support they provided me through my entire life.

CONTENTS

Page

Page

List of Tables

Page

List of Figures

Page

Chapter 1: Introduction

A diversity of computer vision tasks involves assigning appropriate labels to image pixels or superpixels and associated features subjected to spatial and visual contexts. Common applications include image denoising, image inpainting, texture synthesis, object segmentation, object recognition, pose estimation, optical flow and motion estimation, stereo correspondence and multi-view reconstruction. These problems are collectively called image labeling problems and have been of central interest in computer vision and related fields for decades. The label spaces for these problems can vastly vary from one application to another. For instance, the label space for stereo correspondence is the set of possible disparity values for the pixels whereas the label space for object recognition is the set of known objects classes to be inferred from the input image.

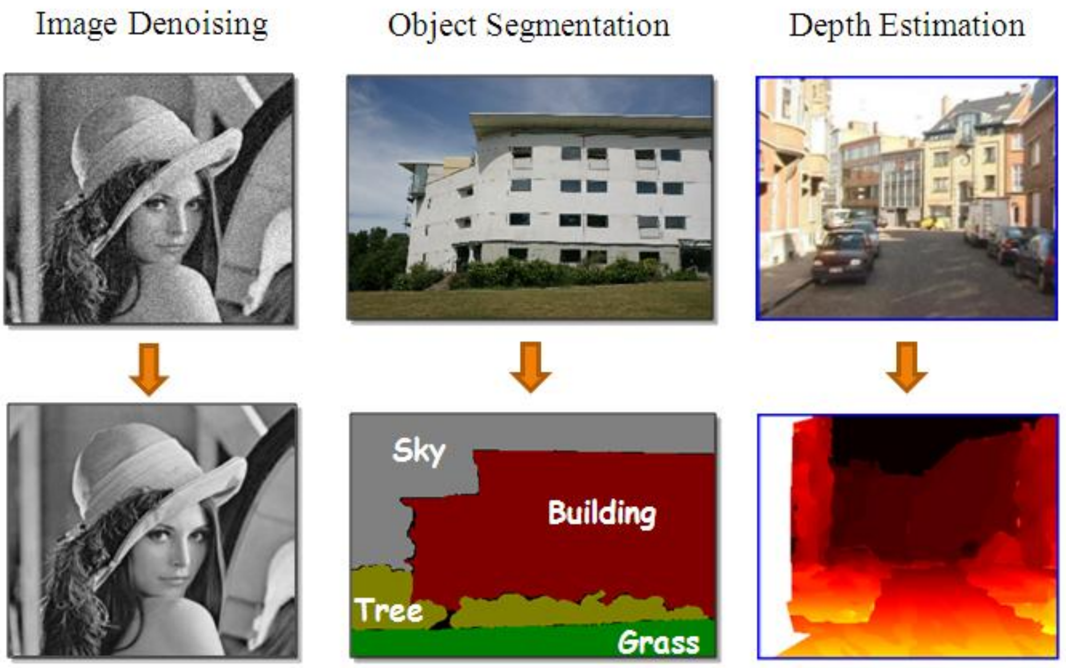
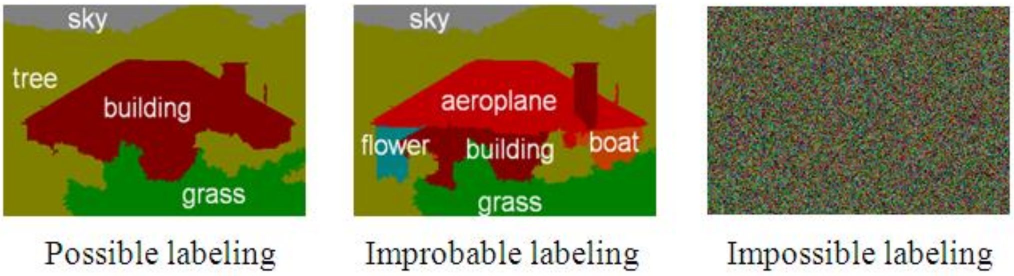


Figure 1: Examples of image labeling problems[1]

[1] Images taken from CVPR 2012 tutorial: Graph Cut based Optimisation for Computer Vision URL: www.robots.ox.ac.uk/~lubor/tutorial.html

Figure 1 shows a few instances of labeling problems. The label space of image denoising is the set of possible intensity values (or color values) and the task is to assign to each pixel its "correct" intensity value based on the given noisy intensity value of that pixel by taking into account the intensity values of its neighbors. For object segmentation, the features of each known object class are learned from training set to construct the model for that object class. These constructed object models are then used to assign labels to the input image superpixels according to their probability of belonging to each object class. In depth estimation from stereo images, the input is a pair of images taken from different viewpoints of the same scene and the task is to assign each pixel its most probable disparity value, which can be used to reconstruct the depth value.

Labeling problems are highly structured for most applications. This is because the labels are spatially correlated with the labels of its neighbors often via complex dependencies specific to each application. Attempting to assign a label to each pixel independently using information from its associated features alone often results in impossible labeling, as exemplified in the right subfigure of Figure 2.[1] Preprocessing by segmenting the input image into superpixels helps improve the result but might still give improbable labeling as shown in the middle subfigure. To obtain the labeling as shown in the left subfigure, the whole task should be formulated as one optimization problem where visual contexts are used to assist in label estimation by providing cues for dependencies between pixel labels.



Possible labeling

Improbable labeling

Impossible labeling

Figure 2: Examples of labeling results for object segmentation

Random field formulation and energy minimization framework provide a natural, elegant, and expressive means of modeling these labeling problems by using a set of random variables to represent image locations and an energy function comprised of potentials that capture contextual information and mutual interaction between them. The problem is then to find the optimal labeling, also referred to as optimal configuration, to an objective energy function subjected to certain constraints. From a probabilistic viewpoint, this problem is equivalent to finding a maximum a posteriori (MAP) solution to a corresponding random field. This method has been shown to give respectable results when compared to other techniques in various applications, but as the number of labels increases, the computational complexity grows fast.

In this work, we exploit characteristics of computer vision tasks to obtain a hierarchical discrete energy minimization algorithm for labeling problems with linear-ordered label space. Instead of using one hierarchy for every variable, our approach builds different local hierarchy for each variable by taking the information from the energy function into account. The added processing steps have significantly less theoretical complexity than the overall process and our algorithm can assist in speeding up the computation time while providing comparable energy as shown in the experimental results.

1.1 Problem Statement

Given a discrete pairwise random field (or, equivalently, a discrete pairwise energy function) on a set of random variables described by a graph, compute a maximum a posteriori estimation.

1.2 Objective of the Work

- To develop an efficient algorithm for optimizing discrete pairwise random fields

Many problems in computer vision, image processing, and related fields can be put in terms of random field or energy function. Even though the formulation in detail lies in the respective applications, the problems reduce to finding a configuration with the minimum energy or solving for a MAP solution. No information concerning the application appears once the formulation is done. Any information that affects the calculation is implicit in the energy function. Our goal is to exploit the characteristics of computer vision tasks to obtain a faster energy minimization algorithm that still gives reasonably good solutions.

1.3 Research Plan

a) Conduct literature survey

b) Identify current trend in the literature and the current problem involving discrete optimization in computer vision

c) Study related works in discrete random field optimization and energy minimization in detail

d) Develop a heuristic approximation algorithm that works faster while still providing good solutions

e) Experiment with computer vision tasks using well-known datasets

f) Perform in-depth analyses on experimental results and examine the proposed heuristic algorithm in contrast with related works

g) Develop a complete algorithm for the problem

h) Perform comparison with other representative algorithms in the literature

i) Publish a journal article relating to the work

j) Prepare and engage in a thesis defense

1.4 Scope of the Work

- This work considers random field optimization and energy minimization in a discrete sense, i.e., the label set is discrete and finite.

- The label space is assumed to have a natural linear ordering structure that represents physical quantity.

- The developed algorithm will be tested against at least three computer vision problems with linearly structured label spaces.

- The developed algorithm will reduce computation time while providing comparable or better energies. Performance measurement can be either actual experimental results or a complexity analysis.

1.5 Contributions

We have proposed a speedup scheme for optimizing discrete pairwise random fields using local label hierarchy. The proposed scheme takes advantage of characteristics of computer vision tasks with linearly ordered label space via a hierarchical energy minimization approach. We give definitions and notations for local label hierarchy and present techniques for label-wise grouping. Definition of energy function is generalized to include sets of labels and heuristics for group potential assignment are discussed. Unlike others, our approach builds different hierarchy for each variable which enables us to achieve better performance compared to using the same hierarchy for every variable despite using the same heuristics to obtain group potentials. Our most competitive technique has a speedup of an order of magnitude with less than 5% increase in energy.

1.6 Benefits of the Work

Obtaining an improved algorithm for optimizing discrete random fields. The algorithm is applicable to many tasks in the area of computer vision and related fields.

1.7 Dissertation Organization

The rest of this dissertation is organized as follows; Chapter 2 provides preliminaries and reviews of related research in the literature concerning random field theory and inference algorithms. The proposed methodology for random field optimization via local hierarchical label-wise grouping is described in Chapter 3. Experiments and results are detailed in Chapter 4. Chapter 5 concludes the thesis.

Chapter 2: Preliminaries and Literature Reviews

Discrete energy minimization problems that are habitually encountered in computer vision applications can be described in the form of random fields. This unified framework offers a consolidated way to separate the energy minimization model from the original labeling problem. This means that any energy minimization algorithm can be used to obtain the solution once the labeling problem is transformed into energy minimization problem. This ability to efficiently port from one optimization algorithm to another enables researchers to choose inference algorithm best suited for their applications of interest.

There are several classes of inference algorithms that attempt to solve discrete energy minimization problems: 1) maximum-flow algorithms based on graph cuts, 2) move-making algorithms that iteratively find labeling with lower energy, 3) algorithms based on message passing, 4) algorithms based on linear programming relaxations, and 5) combinatorial algorithms. Although solving for exact solutions to these problems in general is NP-hard, these classes of algorithms are known to give practical approximate solutions and provide different problem-dependent theoretical guarantees.

This chapter provides a review on random fields as well as several inference algorithms for MAP estimation by optimizing discrete energy minimization problems. The transformations from the original labeling problems into their corresponding energy minimization problems are specific to their respective applications and will be discussed later in Chapter 4.

2.1 Random Fields

A random field can be defined by a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{C})$ with a set of variables $\mathcal{V}$ corresponding to the set of image locations to be labeled, and a set of hyperedges[2] $\mathcal{C}$ characterizing mutual interactions among them. Let $\mathcal{L} = \{l_1, l_2, ..., l_k\}$ be a finite set of possible labels that can be assigned to the variables in $\mathcal{V}$. A configuration

[2] A hyperedge is a non-empty subset of the set of variables. Vertices participating in a hyperedge are said to be neighbors.

$\mathbf{x} = \{x_1, x_2, ..., x_{|\mathcal{V}|}\} \in \mathcal{L}^{|\mathcal{V}|}$ of the random field is a labeling of the variables and has an energy value defined as

$$E(\mathbf{x}) = \theta_{const} + \sum_{v \in \mathcal{V}} \theta_v(x_v) + \sum_{c \in \mathcal{C}} \theta_c(\mathbf{x}_c) \tag{2.1}$$

where each unary potential $\theta_v : \mathcal{L} \to \mathbb{R}$ is a function that outputs the energy of assigning label x_v to a variable v and each higher-order potential $\theta_c : \mathcal{L}^{|c|} \to \mathbb{R}$ represents the energy of assigning labels $\mathbf{x}_c = \{x_v \mid v \in c\}$ to a set of variables $\mathbf{v}_c$ participating in hyperedge c. Figure 3 shows an example of a random field with $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{C} = \{c_1, c_2, c_3\} = \{\{v_1, v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}\}$. In this example, the associated energy function is $E(\mathbf{x}) = \theta_{const} + \theta_{v_1}(x_1) + \theta_{v_2}(x_2) + \theta_{v_3}(x_3) + \theta_{v_4}(x_4) + \theta_{c_1}(x_1, x_2, x_3) + \theta_{c_2}(x_2, x_4)$ $+\theta_{c_3}(x_3, x_4)$.[3]

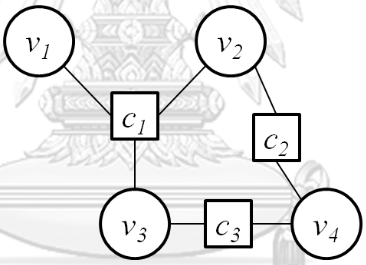


Figure 3: Example of a random field with four variables and three hyperedges

For a random field, we seek an optimal configuration $\mathbf{x}^*$ that minimize the energy function $E(\mathbf{x})$ or, equivalently, maximize the probability distribution $\Pr(\mathbf{x}) = \exp(-E(\mathbf{x}))/Z$ of the random field where $Z = \sum_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} \exp(-E(\mathbf{x}))$ is the partition function. Note that the probability distribution satisfies the Markov property: Every variable is conditionally independent of all other variables given its neighbors,

[3] The curly braces in $\theta_c(\cdot)$ are omitted for better readability.

i.e., for every random variable v, $\Pr(x_v \mid \mathbf{x}_{\mathcal{V}-\{v\}}) = \Pr(x_v \mid \mathbf{x}_{\mathcal{N}_v})$ where $\mathcal{N}_v$ is the set of neighbors of a variable v.

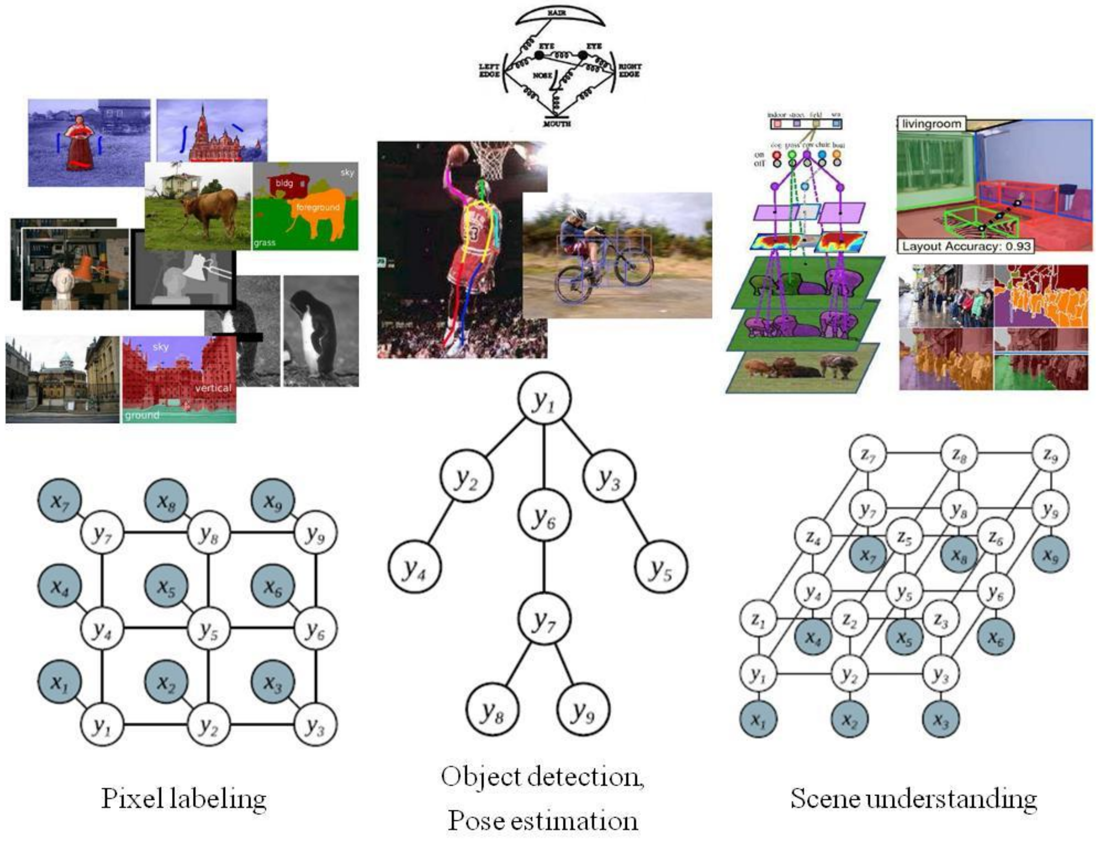
Strictly speaking, the above definition is for a Markov random field (MRF) [Kindermann and Snell 1980]. In cases where some variables already have labels, using MRFs to find the MAP solution for all variables may lead to overly-involved calculations. Conditional random fields (CRF) [Lafferty *et al.* 2001] do not include the dependencies between these variables in the calculation and choose to model the probability distribution conditioned on these observed variables instead. Formally, a CRF is a hypergraph together with a probability distribution such that every variable $v \in \mathcal{V}$ obeys the Markov property with respect to $\mathcal{G}$ when given the labels of observed variables.

For problems in computer vision, it is often the case that the observed variables of conditional random field are given so that optimizing a CRF reduces to optimizing an MRF. Due to their pervasive uses in computer vision community, we focus on MRFs in our work.

Figure 4 shows some examples of random field formulation for different classes of applications in computer vision. Pixel-labeling problems (left) such as foreground/background estimation, dense stereo correspondence, and image segmentation require a label to be assigned to each pixel separately. To accomplish this, a variable in the model is created to be associated with each pixel in the input image, often referred to as pixel-based models.

For the tasks involving object detection and pose estimation (middle), input images are often preprocessed into superpixels before assigning labels, thus forcing a single label to be assigned to all pixels in the same superpixel. The main reason for using superpixel is that objects can hardly ever, if possible at all, be recognized from a single pixel. The output of object recognizer often provides scores for how likely it is for the input superpixel to belong in the known object classes. The mutual interactions in the hyperedges between the superpixel variables can then help incorporate the spatial relationships between objects, which make label assignment much more probable.

For scene understanding (right), more layers are often needed to assemble pixels into progressively larger parts. The model, therefore, often consists of several levels from fine to coarse with both intralevel and interlevel interactions. These problems can also require each pixel to be labeled independently but, compared to pixel-labeling problems, scene understanding problems require much more complex mutual interactions to accomplish the task, which can even lead to the model being intractable in some cases.


HAIR
EYE
EYE
LEFT EDGE
NOSE
RIGHT EDGE
MOUTH
sky
bldg
foreground
grass
sky
vertical
ground
livingroom
Layout Accuracy: 0.93
x_7 x_8 x_9 y_7 y_8 y_9
x_4 x_5 x_6 y_4 y_5 y_6
x_1 x_2 x_3 y_1 y_2 y_3
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9
z_7 z_8 z_9 z_4 z_5 z_6 z_1 z_2 z_3

Pixel labeling

Object detection, Pose estimation

Scene understanding

Figure 4: Examples of random field formulation in vision applications[4]

2.2 Maximum-Flow Algorithms

Before we delve into the matter of how graph-based energy minimization is done, it is worthwhile to first introduce the concept of graph cuts. Let $G = (V, E)$ be a directed

[4] Images taken from CVPR 2014 Tutorial on Learning and Inference in Discrete Graphical Models URL: http://imagine.enpc.fr/~komodakn/GraphicalModels_CVPR2014.html

weighted graph where the vertex set V contains two special vertices, the source s and the sink t, and each directed edge (p,q) in the edge set $E \subseteq V \times V$ has a non-negative weight (also referred to as capacity) denoted by $c(p,q)$. An s-t cut is a partition of V into two disjoint subsets S and T with $s \in S$ and $t \in T$. The cost of an s-t cut is defined by

$$c(S,T) = \sum_{(p,q)\in E,\, p\in S,\, q\in T} c(p,q) \tag{2.2}$$

and the minimum s-t cut problem is the task of finding an s-t cut that has minimum cost. We will refer to an s-t cut as simply a "cut" from this point onward for the sake of simplicity.

Assigning vertices in S and T with label 0 and 1, respectively, allows a cut to be reinterpreted in terms of energy minimization as a binary configuration. Constructing a suitable graph such that each of its cut corresponds to an energy configuration (and vice versa) means that its minimum cut would correspond to the minimum energy configuration and can be used as the solution. An assortment of algorithms with known polynomial-time can then be used to calculate a maximum flow [Ahuja *et al.* 1993, Goldberg 1998] and, by the max-flow/min-cut theorem [Ford and Fulkerson 1956], a minimum cut of the graph.

Table 1 shows a list of maximum flow algorithms [Goldberg 1998]. The years refer to the first publication of each algorithm. In the table, F denotes the maximum capacity in the graph and O^* denotes the expected running time of a randomized algorithm. From the table, two main classes of the algorithms exist: push-relabel style algorithms (P) and augmenting-path-based algorithms (A). For computer vision tasks, the constructed graphs are most commonly in the form of two or higher dimensional grid [Boykov and Veksler 2006]. In this case, the augmenting path algorithm in [Boykov and Kolmogorov 2004] has been shown to achieve near-linear running time for many vision applications, even though the authors do not have a polynomial bound for the algorithm.

Table 1: List of max-flow algorithms

Year	**Representative work**	**Complexity**
1951	[Dantzig 1951]	$O(\vert V\vert^2\vert E\vert F)$
1956	[Ford and Fulkerson 1956] (A)	$O(\vert V\vert\vert E\vert F)$
1970	[Dinic 1970] (A)	$O(\vert V\vert\vert E\vert^2)$, $O(\vert V\vert^2\vert E\vert)$
1972	[Edmonds and Karp 1972] (A)	$O(\vert V\vert\vert E\vert^2)$, $O(\vert E\vert^2 \log F)$
1973	[Dinic 1973] (A)	$O(\vert V\vert\vert E\vert \log F)$, $O(\vert E\vert^2 \log F)$
1974	[Karzanov 1974] (A)	$O(\vert V\vert^3)$
1977	[Cherkassky 1977] (A)	$O(\vert V\vert^2 \sqrt{\vert E\vert})$
1980	[Galil and Naamad 1980] (A)	$O(\vert V\vert\vert E\vert \log^2 \vert V\vert)$
1983	[Sleator and Endre Tarjan 1983] (A)	$O(\vert V\vert\vert E\vert \log \vert V\vert)$
1986	[Goldberg and Tarjan 1986] (P)	$O(\vert V\vert\vert E\vert \log(\vert V\vert^2 / \vert E\vert))$
1987	[Ahuja and Orlin 1989] (P)	$O(\vert V\vert\vert E\vert + \vert V\vert^2 \log F)$
1987	[Ahuja *et al.* 1989] (P)	$O(\vert V\vert\vert E\vert \log(\vert V\vert \sqrt{\log F} / \vert E\vert))$
1989	[Cheriyan and Hagerup 1989, Cheriyan and Hagerup 1995] (P)	$O^*(\vert V\vert\vert E\vert + \vert V\vert^2 \log^2 \vert V\vert)$
1990	[Cheriyan *et al.* 1996] (P)	$O(\vert V\vert^3 / \log \vert V\vert)$
1990	[Alon 1990] (P)	$O(\vert V\vert\vert E\vert + \vert V\vert^{8/3} \log \vert V\vert)$
1992	[King *et al.* 1992] (P)	$O(\vert V\vert\vert E\vert + \vert V\vert^{2+\epsilon})$
1993	[Phillips and Westbrook 1993] (P)	$O(\vert V\vert\vert E\vert (\log_{\vert E\vert/\vert V\vert} \vert V\vert + \log^{2+\epsilon} \vert V\vert))$
1994	[King *et al.* 1994] (P)	$O(\vert V\vert\vert E\vert \log_{\vert E\vert/(\vert V\vert\log\vert V\vert)} \vert V\vert)$
1997	[Goldberg and Rao 1997, Goldberg and Rao 1998] (A)	$O(\min(\vert V\vert^{2/3}, \sqrt{\vert E\vert}) \vert E\vert \log\left(\frac{\vert V\vert^2}{\vert E\vert}\right) \log F)$

In first-order (pairwise) energy function where the hyperedges are limited to cardinality of two, the energy function of a configuration $\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}$ is of the form

$$E(\mathbf{x}) = \theta_{const} + \sum_{p \in \mathcal{V}} \theta_p(x_p) + \sum_{pq \in \mathcal{E}} \theta_{pq}(x_p, x_q) \tag{2.3}$$

where each θ_p is the unary potential of a variable p and θ_{pq} is the pairwise potential of an undirected edge $\{p, q\}$. Note that θ_{pq} and θ_{qp} refer to the same potential and, therefore, appear only once in the energy function.

2.2.1 Binary Energy Minimization based on Maximum-Flow

If the labels are from the Boolean set $\{0,1\}$, this form of energy is called quadratic pseudo-Boolean function in view of the fact that it can be rewritten as a quadratic polynomial and it maps Boolean variables to the set of real numbers $\mathbb{R}$ instead of the Boolean set.

Calculating the minimum cut of a suitable graph can give exact solution if the binary energy function of the form in Eq. (2.3) is submodular [Kolmogorov and Zabih 2004], i.e., if all its pairwise terms satisfy

$$\theta_{pq}(0,0) + \theta_{pq}(1,1) \leq \theta_{pq}(0,1) + \theta_{pq}(1,0). \tag{2.4}$$

This submodular condition guarantees that only graphs with non-negative capacities can be constructed so that an associated maximum flow can be calculated in polynomial time. Although this is a restrictive class of energy functions, several influential algorithms have been proposed in the literature and provided notable results in many applications. For general graphs (with the possibility of negative capacities), unfortunately, the problem is proven to be NP-hard since negating the capacities allows the maximum cut problem (one of Karp's 21 NP-complete problems [Karp 1972]) to be reduced to it. Furthermore, the maximum flow problem is P-complete[5] and, therefore,

[5] A problem is P-complete if it is in P (a deterministic Turing machine can solve it in polynomial time) and every problem in P can be reduced to it.

is believed to be inherently sequential[6] [Goldschlager *et al.* 1982, Greenlaw *et al.* 1995, Nieuwenhuis *et al.* 2013].

For simplicity, let $\theta_{p,i}$ denote $\theta_p(i)$ and $\theta_{pq,ij}$ denote $\theta_{pq}(i,j)$ where $i,j \in \{0,1\}$ and let $\theta_p = \{\theta_{p,0}, \theta_{p,1}\}$ and $\theta_{pq} = \{\theta_{pq,00}, \theta_{pq,01}, \theta_{pq,10}, \theta_{pq,11}\}$ be vectors of size two and four, respectively. The parameter vector $\boldsymbol{\theta} = \{\theta_\alpha \mid \alpha \in \mathcal{I}\}$ denotes the concatenation of all the terms in the energy function into a single vector where

$$\mathcal{I} = \{(p,i) \mid p \in \mathcal{V} \wedge i \in \{0,1\}\} \cup \{(pq,ij) \mid p,q \in \mathcal{V} \wedge i,j \in \{0,1\}\} \cup \{const\}$$

is the index set. This parameter vector $\boldsymbol{\theta}$ can be used to completely specify the binary energy function in Eq. (2.3), which will be denoted by $E(\mathbf{x} \mid \boldsymbol{\theta})$. Note that the parameter vector of an energy function is not unique. For example, one can obtain the same energy function by subtracting a constant from both entries of θ_p and adding the same constant to θ_{const}. A parameter vector $\boldsymbol{\theta}'$ is called a reparametrization of a parameter vector $\boldsymbol{\theta}$ if, for every configuration $\mathbf{x}$, $E(\mathbf{x} \mid \boldsymbol{\theta}) = E(\mathbf{x} \mid \boldsymbol{\theta}')$.

A parameter vector $\boldsymbol{\theta}$ is said to be in a normal form the following two conditions are met:

a) $\min(\theta_{p,0}, \theta_{p,1}) = 0$ for all $p \in \mathcal{V}$ and

b) $\min(\theta_{pq,00}, \theta_{pq,10}) = 0$ and $\min(\theta_{pq,01}, \theta_{pq,11}) = 0$ for all $\{p,q\} \in \mathcal{E}$.

The following reparametrization procedure [Kolmogorov and Rother 2007] provides how to obtain a normal form.

1) While there exists an edge $\{p,q\}$ and a label $j \in \{0,1\}$ not satisfying Condition b): Calculate $\delta = \min(\theta_{pq,0j}, \theta_{pq,1j})$, then subtract δ from $\theta_{pq,0j}$ and $\theta_{pq,1j}$ and add δ to $\theta_{q,j}$.

[6] Otherwise it would imply that every problem in P can be efficiently solved in a parallel manner.

2) For every vertex p: calculate $\delta = \min(\theta_{p,0}, \theta_{p,1})$, subtract δ from $\theta_{p,0}$ and $\theta_{p,1}$ and add δ to θ_{const}.

Note that either $\theta_{pq,00} = \theta_{pq,11} = 0$ (the pairwise term is submodular) or $\theta_{pq,01} = \theta_{pq,10} = 0$ (the pairwise term is supermodular) for each edge $\{p, q\}$ upon termination.

We first describe an algorithm where the energy is submodular. After parameter reparametrization into a normal form, a directed weighted graph $G = (V, E)$ is created by setting $V = \mathcal{V} \cup \{s, t\}$. The edge capacities are then set such that $c(s, p) = \theta_{p,1}$, $c(p, t) = \theta_{p,0}$, $c(p, q) = \theta_{pq,01}$ and $c(q, p) = \theta_{pq,10}$[7] when $\theta_{\alpha} \neq 0$. It can be proven that every configuration has an energy equal to the cost of the corresponding cut in this graph plus θ_{const}. The construction is shown in Figure 5 (a). One can see, for example, that the cut for the configuration $x_p = 0$ and $x_q = 1$ includes the directed edges $\theta_{p,0}$, $\theta_{q,1}$ and $\theta_{pq,01}$ as shown in Figure 5 (b), which, together with θ_{const}, sum up to the correct amount of energy.

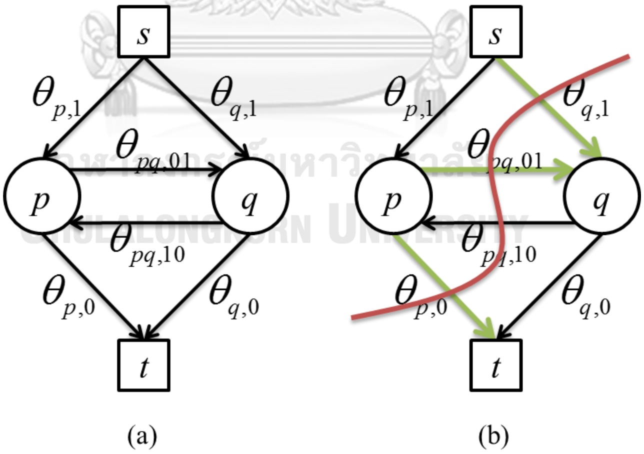


Figure 5: Graph construction for submodular energy function

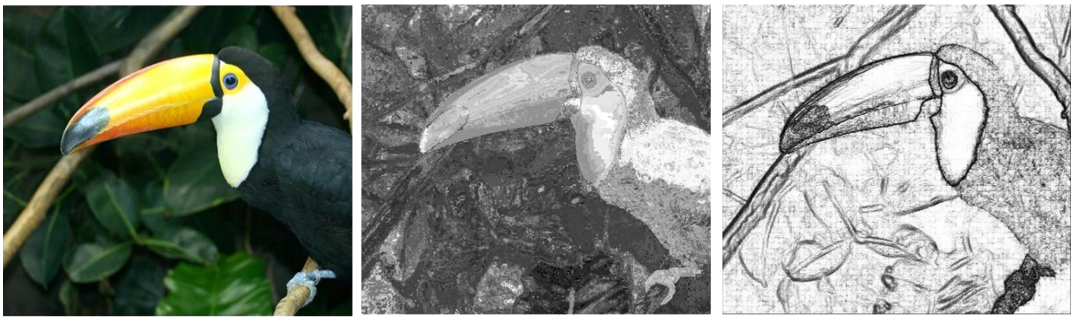
[7] This last case is the same as setting $c(q, p) = \theta_{qp,01}$ and is added only for clearer illustration.

The algorithm then computes maximum flow of the constructed graph. We give an example of using an augmenting-path-based max-flow computation. An augmenting path is a term given for a path from the source s to the sink t through the graph in which every directed edge has positive capacity. While there exists an augmenting path, let δ be the minimum capacity of the edges along the path, subtract δ from the capacities of all edges in the path (including terminal edges[8]) and add δ to the capacities of the corresponding reverse edges in the path (there is no edge going into s and going out of t) and to θ_{const}. This procedure (usually called "pushing" flow) is a form of reparametrization and results in a less complicated normal form where the capacities of the terminal edges in the path are always decreased. The procedure terminates when no more augmenting path exists and the resulting θ_{const} is the optimal energy [Ford and Fulkerson 1956].

After maximum flow computation, any vertex that is reachable by s is assigned label 0 and any vertex from which t is reachable is assigned label 1. Other vertices can be assigned arbitrarily without changing the energy provided that the labels are consistent within each connected component. It is worth to remark that this graph-based energy minimization algorithm for submodular energy function is equivalent to max-product belief propagation with appropriate scheduling and damping scheme [Tarlow *et al.* 2011].

Figure 6 shows an example of binary energy function constructed from the task of foreground/background estimation. The unary potential in this case is the likelihood of each pixel to be in the background or foreground classes computed from the color model of each class (brighter value indicates more likely to be foreground). The pairwise potential reflects the likelihood of neighboring pixels to have the same label, i.e., to both belong to foreground (or background), and is calculated from pixel discontinuity (brighter value indicates weaker image edge or more likely to belong to the same class).

[8] A terminal edge is an edge that is connected to either s or t.



Input image

Unary potential

Pairwise potential

Figure 6: Binary energy minimization in foreground/background estimation[9]

Several works involving non-submodular energy functions employ certain conversion to ensure that all terms are submodular prior to max-flow computation. For some tasks, truncating [Rother *et al.* 2005] or ignoring [Cremers and Grady 2006] non-submodular terms provide reasonable results. For other applications, exclusion of non-submodular terms can decrease the quality of the results [Kolmogorov and Rother 2007], especially for models with substantial numbers of them.

For quite a while, QPBO (quadratic pseudo-Boolean optimization) method [Boros and Hammer 2002] has grown sizeable attention in computer vision community. This method uses two vertices p and $\bar{p}$ for each variable $p \in \mathcal{V}$ instead of using one vertex per one binary variable to cope with supermodular terms. For any non-submodular term $\theta_{pq}(x_p, x_q)$ with

$$\theta_{pq}(0,0) + \theta_{pq}(1,1) > \theta_{pq}(0,1) + \theta_{pq}(1,0),$$

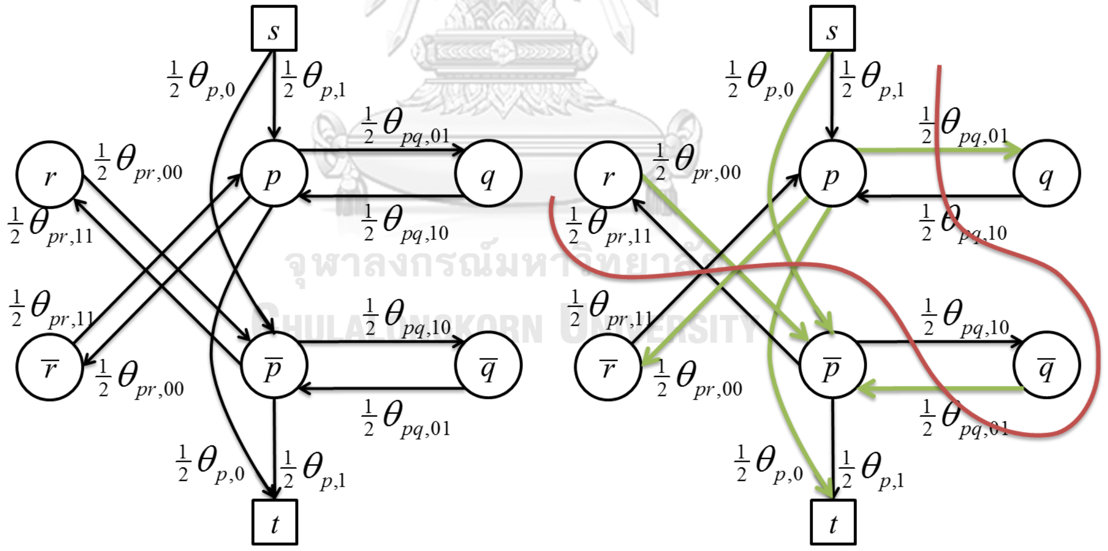
replacing q with $\bar{q}$ results in

$$\theta_{p\bar{q}}(0,1) + \theta_{p\bar{q}}(1,0) > \theta_{p\bar{q}}(0,0) + \theta_{p\bar{q}}(1,1)$$

which is submodular and, therefore, can be converted into min-cut graph and solved in polynomial time.

[9] Images taken from ECCV 2008 MAP Estimation Algorithms in Computer Vision – Part II URL: http://www.robots.ox.ac.uk/~pawan/eccv08_tutorial/index.html

By applying the technique based on this observation (known as roof dual relaxation [Hammer *et al.* 1984]), QPBO method is able to minimize any first-order binary energy function of the form in Eq. (2.3) in polynomial time. Similar to the submodular case, a graph $G=(V,E)$ is constructed, but this time with $V=\{p,\bar{p} \mid p\in\mathcal{V}\}\cup\{s,t\}$. The edge capacities are set such that $c(p,t)=c(s,\bar{p})=\frac{1}{2}\theta_{p,0}$, $c(s,p)=c(\bar{p},t)=\frac{1}{2}\theta_{p,1}$ for each unary potential $\theta_p\neq 0$ and $c(p,\bar{q})=c(q,\bar{p})=\frac{1}{2}\theta_{pq,00}$, $c(p,q)=c(\bar{q},\bar{p})=\frac{1}{2}\theta_{pq,01}$, $c(q,p)=c(\bar{p},\bar{q})=\frac{1}{2}\theta_{pq,10}$, $c(\bar{p},q)=c(\bar{q},p)=\frac{1}{2}\theta_{pq,11}$ for pairwise potential $\theta_{pq}\neq 0$. The construction is shown in Figure 7 (a). In the figure, the pairwise term for the edge $\{p,q\}\in\mathcal{E}$ is submodular ($\theta_{pq,00}=\theta_{pq,11}=0$) whereas the pairwise term for edge $\{p,r\}$ is supermodular ($\theta_{pr,01}=\theta_{pr,10}=0$). The edges corresponding to the unary potentials of q and r are not shown for better readability. Figure 7 (b) shows the cut for the configuration $x_p=x_r=0$ and $x_q=1$.



s
$\frac{1}{2}\theta_{p,0}$
$\frac{1}{2}\theta_{p,1}$
$\frac{1}{2}\theta_{pq,01}$
$\frac{1}{2}\theta_{pr,00}$
r
p
q
$\frac{1}{2}\theta_{pr,11}$
$\frac{1}{2}\theta_{pq,10}$
$\frac{1}{2}\theta_{pr,11}$
$\frac{1}{2}\theta_{pq,10}$
$\bar{r}$
$\bar{p}$
$\bar{q}$
$\frac{1}{2}\theta_{pr,00}$
$\frac{1}{2}\theta_{pq,01}$
$\frac{1}{2}\theta_{p,0}$
$\frac{1}{2}\theta_{p,1}$
t

(a) (b)

Figure 7: Graph construction for QPBO method

A maximum flow is then computed and the vertices in V are assigned with labels from the corresponding minimum cut in the same manner as before. However, relaxing the integrality constraint (x_p and $x_{\bar{p}}$ are not constrained to be complements of each other's) means that QPBO method can output a partial configuration $\mathbf{x} \in \{0,1,\varnothing\}^{|\mathcal{V}|}$ where $x_p \in \{0,1\}$ if p and $\bar{p}$ "agree" ($x_{\bar{p}} = 1 - x_p$) and $x_p = \varnothing$ (unlabeled) otherwise. It is guaranteed by the partial optimality property [Boros and Hammer 2002, Hammer, *et al.* 1984] that the output partial configuration is part of some global optimal solution. In other words, there exists a global optimal configuration $\mathbf{x}^*$ such that $x_p^* = x_p$ for every labeled variable p.

2.2.2 Multi-label Energy Minimization based on Maximum-Flow

One rather intuitive way to handle energy minimization in the multi-label case is to convert it into multi-terminal cut problem. Given a weighted graph with k terminals, the problem is to find a set of separating edges with minimum cost that partition the vertex set into k unconnected subsets with each terminal in each of these subsets (this problem reduces to an s-t minimum-cut problem when k is two). Unfortunately, the problem has been proven to be NP-hard for general graphs once k is greater than two [Dahlhaus *et al.* 1994]. Approximation algorithms for this problem can be found in [Boykov *et al.* 1998, Karger *et al.* 1999, Xiao 2008].

Graph-based multi-label energy minimization algorithms can be categorized into two broad classes. One class, collectively known as move-making algorithms, iteratively finds local optimum through possible labels whereas the other class attempts to calculate the global optimum using all labels at once.

Move-making algorithms in general maintain a labeling on hand. A new labeling is proposed in each iteration and each vertex collaboratively decides whether to retain its old label or move to the one proposed in that iteration. For computer vision tasks, this class of algorithms was first popularized by the work of [Boykov *et al.* 2001] which was based on maximum flow. Move-making algorithms will be discussed in detail in section 2.3.

Early works that attempt to compute all labels at once [Roy 1999, Thomo *et al.* 1998, Zhao 2000] use push-relabel style algorithms [Goldberg and Tarjan 1986] to minimize energy. However, only a limited range of pairwise interactions can be used to allow for the max-flow routine to be calculated in polynomial time. For energy functions with convex priors[10], the algorithm in [Ishikawa 2003] is proven to compute exact solutions. Despite the ability to solve for a global minimum exactly and efficiently, the results from having convex prior are often over-smoothed due to it not being discontinuity preserving [Kolmogorov and Zabih 2004].

To circumvent the limitation of the underlying max-flow that requires the energy to be submodular, a multi-label variant of QPBO method (MQPBO) has been introduced [Kohli *et al.* 2008]. MQPBO allows for a broader class of energy functions to be minimized at the cost of allowing "undecided" label in the solutions. A partial solution calculated by MQPBO is in the form of an interval for each variable. These intervals together are proven to contain an optimal solution, which allows for labels outside of them to be discarded.

Given a multi-label energy function, MQPBO method constructs a corresponding binary energy function[11] as follows. In this case, the label set $\mathcal{L}$ is presumed to be in a linear order structure, i.e., $\mathcal{L} = \{0,1,\ldots,k-1\}$. The set of variables is defined as $V = \{(p,d) : p \in \mathcal{V} \wedge d \in \mathcal{L} - \{0\}\} \cup \{s,t\}$, corresponding to the discrete solution space defined by the random variables and possible labels.

If a variable p were to have label d, the corresponding binary configuration would be for a vertex (p,i) to have value 1 if $i \le d$, and 0 otherwise. To ensure that this would be the case, hard constraints are added in the form of pairwise potential $\Theta_{(p,d-1),(p,d)}(0,1) = \infty$ to every pair of vertices (p,d) and $(p,d-1)$ with $d \ge 2$. The unary potential of each vertex (p,d) with $d \ge 1$ is set as

$$\begin{aligned}\Theta_{(p,d)}(1) = \theta_p(d) - \theta_p(d-1) + \sum_{q:pq\in\mathcal{E}} (\theta_{pq}(d,0) + \theta_{pq}(0,d)) \\ - \sum_{q:pq\in\mathcal{E}} (\theta_{pq}(d-1,0) + \theta_{pq}(0,d-1)).\end{aligned} \tag{2.5}$$

[10] The label set $\mathcal{L}$ can be placed in a linear order such that every pairwise term $\theta_{pq}(x_p, x_q) = \alpha_{pq} g(x_p - x_q)$ where g is a convex function.

[11] For simplicity, we chose to explain MQPBO in terms of energy functions. The maximum-flow graph is constructed according to the same rules as QPBO.

The pairwise potential for each pair of vertices $(p,i),(q,j)$ with $pq \in \mathcal{E}$ and $i,j \in \mathcal{L}$ are set as

$$\Theta_{(p,i),(q,j)}(1,1) = \theta_{pq}(i,j) + \theta_{pq}(i-1,j-1) - \theta_{pq}(i,j-1) - \theta_{pq}(i-1,j). \qquad (2.6)$$

Finally, set

$$\Theta_{const} = \sum_{p\in\mathcal{V}} \theta_p(0) + \sum_{pq\in\mathcal{E}} \theta_{pq}(0,0) + \theta_{const} \qquad (2.7).$$

It can be shown [Shekhovtsov *et al.* 2008] that this constructed binary energy function is equivalent to the multi-label energy function for every configuration $\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}$. Note that the construction described above is subject to the ordering of the label set $\mathcal{L}$. Figure 8 (a) shows a simplified graph construction for MQPBO method where the complement vertices are not shown and only a number of edges are present. Figure 8 (b) shows the cut for the configuration $x_p = 0$ and $x_q = 2$.

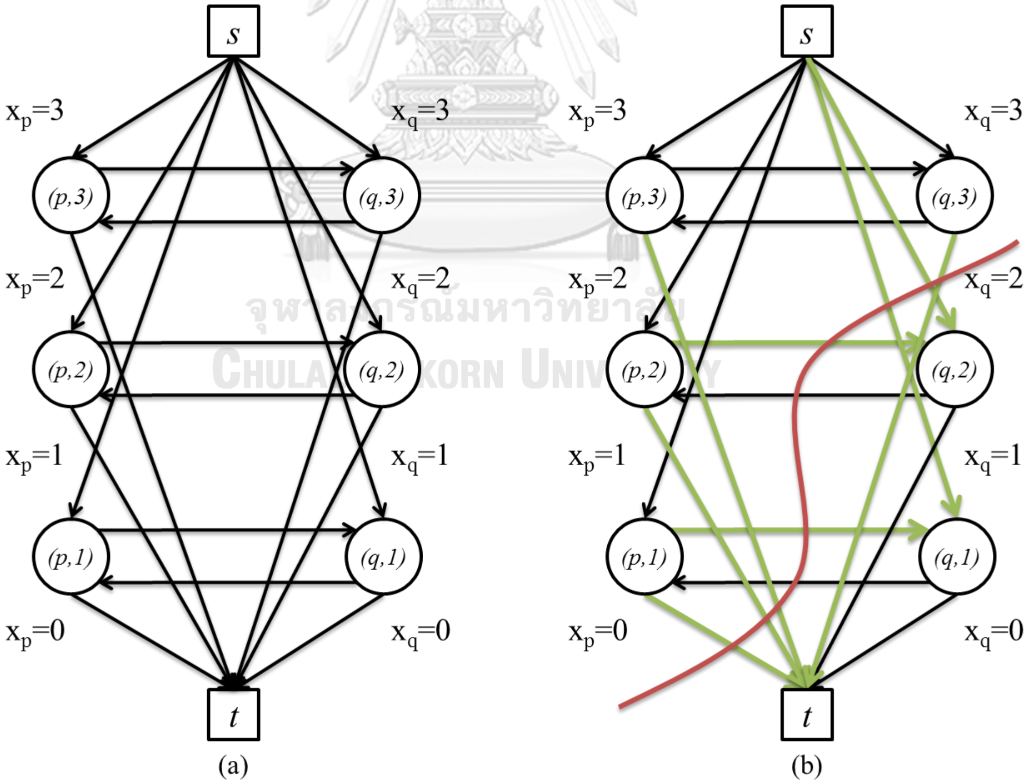


Figure 8: Graph construction for MQPBO method

The ability to handle multi-label energy functions while also not restricting the class of energy being minimized has made MQPBO been our main choice of optimization algorithm in our experiments. One drawback of the multi-label construction is that the size of the set of vertices V is $O(|\mathcal{V}||\mathcal{L}|)$, which, when combined with the complexity of max-flow, can lead to large running time.

2.3 Move-Making Algorithms

Move-making algorithms iteratively find new labeling with lower energy by combining two or more sub-optimal labelings in each iteration. The most prominent algorithm of iterating through the labels is the α-expansion algorithm introduced in [Boykov, *et al.* 2001]. In each iteration, each variable is given a choice to decide whether to keep its original label or "move" to a new label α, hence the name α-expansion. This pioneer algorithm has been established to be very competent in calculating strong local optima for metric[12] energy functions and has been used to provide respectable results by various works [Boykov and Funka-Lea 2006, Boykov and Kolmogorov 2004, Felzenszwalb and Huttenlocher 2005, Hirschmuller 2005, Kim *et al.* 2003, Kolmogorov and Zabih 2001, Kolmogorov and Zabih 2002, Kwatra *et al.* 2003, Scharstein and Szeliski 2002].

The fact that graph cuts can find exact solutions to submodular energy functions in the binary case poses a limitation to α-expansion, i.e., it can only find approximate solutions of metric energy functions. The authors of [Boykov, *et al.* 2001] also proposed another algorithm, the $\alpha\beta$-swap, in which some variables with label α swap to label β and vice versa. The graph construction of $\alpha\beta$-swap allows semi-metric energy functions to be used, but without the optimality guarantee of α-expansion.

[12] A multi-label energy function is called metric if for every $\{p,q\} \in \mathcal{E}$ and $\alpha,\beta,\gamma \in \mathcal{L}$, its pairwise terms satisfy: a) $\theta_{pq}(\alpha,\beta)=0 \Leftrightarrow \alpha=\beta$, b) $\theta_{pq}(\alpha,\beta)=\theta_{pq}(\beta,\alpha)\geq 0$, and c) $\theta_{pq}(\alpha,\beta)\leq\theta_{pq}(\alpha,\gamma)+\theta_{pq}(\gamma,\beta)$. A semi-metric energy function is one whose pairwise terms satisfy a) and b).

Fusion move [Lempitsky *et al.* 2010] employs QPBO method to fuse together two candidate solutions. The α-expansion algorithm can be considered as a specific case where one of the candidate solutions consists entirely of one label and the energy function is metric. The advantage of QPBO is that it allows a broader class of energy function to be handled by their scheme. The use of QPBO was also studied independently in [Woodford *et al.* 2009] and both works have shown promising experimental results.

Several other move-making algorithms exist. LogCut algorithm [Lempitsky *et al.* 2007] encodes the label set as a sequence of bits and uses the binary hierarchy to make decisions starting from the most significant bit. The hierarchical move-making strategy in [Kumar and Koller 2009] uses a mixture of tree metrics to estimate semi-metric energy function and combines the solutions using α-expansion.

Range expansion and range-swap algorithms [Kumar *et al.* 2011, Veksler 2012] extends α-expansion and $\alpha\beta$-swap by enabling the pixels to explore a range of consecutive labels instead of just one, which results in improved multiplicative bound for truncated convex models. GRSA, which was proposed in [Liu *et al.* 2015], explores the idea further and uses intervals with gaps that satisfy submodular condition in range swap move.

Move-making algorithms in general search only a portion of the solution space. The search space size of α-expansion and fusion move, for instance, is $2^{|\mathcal{V}|}$ whereas the entire solution space is $|\mathcal{L}|^{|\mathcal{V}|}$.

FastPD [Komodakis and Tziritas 2007] and Iterated Conditional Modes (ICM) [Besag 1986] can be considered as move-making algorithms. Starting with an estimate of the solution, ICM iteratively applies a greedy winner-take-all strategy to assign the label that provides the maximum decrease in energy for each variable. The process is guaranteed to converge and does so rather fast in practice but, unfortunately, the qualities of the computed solutions highly depend on the initial estimate. The Lazy Flipper [Andres *et al.* 2012] algorithm is analogous to ICM in that it repeatedly uses a greedy exhaustive search without using max-flow methods but over local subsets of variables of size k instead of ICM's one-at-a-time. FastPD is similar to move-making algorithm, specifically α-expansion, in the way it selects and evaluates moves, but FastPD also uses the dual solution of the maximum flow to reparametrize the objective function which can provide considerable speedup but, unfortunately, can get stuck in suboptimal fix-points.

2.4 Message-Passing Algorithms

Message-passing algorithm, or belief propagation, relies on local message updates to compute an optimal labeling. There are two popular variants of message-passing algorithms: sum-product and max-product message passing. Sum-product message passing is used for inference on graphical models. The objective of inference is to compute the marginal distribution, i.e., the probability distribution of every configuration of the unobserved variables conditioned on the observed variables. On the other hand, max-product message passing is used for finding a MAP solution, i.e., a configuration with maximum probability.

Each message contains the values of "belief" that other variables have over each label that a variable v can take. The algorithm uses distributive property over maximum and product operations to break down global optimization into local messages between variables and hyperedges. For max-product message passing, the message $m_{v\to c}(x_v)$ from a variable v to hyperedge c is defined as

$$m_{v\to c}(x_v) = \psi_v(x_v) \prod_{c^*: v\in c^*, c^*\neq c} m_{c^*\to v}(x_v) \tag{2.8}$$

and the message $m_{c\to v}(x_v)$ from c to v is

$$m_{c\to v}(x_v) = \max_{\mathbf{x}_c^*: x_v^* = x_v} \left\{ \psi_c(\mathbf{x}_c^*) \prod_{v^*: v^*\in c, v^*\neq v} m_{v^*\to c}(x_{v^*}^*) \right\} \tag{2.9}$$

where $\psi_v(x_v) = \exp(-\theta_v(x_v))$ and $\psi_c(\mathbf{x}_c) = \exp(-\theta_c(\mathbf{x}_c))$ are shown to emphasize the term "max-product". However, min-sum algorithm is used with $\min\{\cdot\}$, Σ, $\theta(\cdot)$ in places of $\max\{\cdot\}$, Π, $\psi(\cdot)$ for efficiency in implementation. Sum-product messages are defined by replacing $\max\{\cdot\}$ with Σ in Eq. (2.9).

The algorithm converges to a global optimum in two passes of appropriate scheduling if the underlying graph is a tree. For arbitrary graphs with loops, the problem is known to be NP-hard [Weiss and Freeman 2001, Yedidia *et al.* 2003]. Repeatedly applying message passing to general graph is called Loopy Belief Propagation (LBP) [Bishop 2006]. LBP is not guaranteed to converge or give the correct solution in case it does converge. However, it still provides satisfactory results in some cases.

Tree-reweighted message passing (TRW) [Wainwright *et al.* 2005] copes with loops by breaking the graph to be computed into a set of spanning trees that together cover every edge in the original graph and then trying to find a configuration from optimal solutions of the trees. The configuration obtained from the trees is guaranteed to be optimal if the tree solutions satisfy the tree-agreement condition, i.e. the labels for each vertex in every tree are equal. It is not surprising that this condition is not always achieved which means that TRW does not always converge. Sequential Tree-reweighted message passing (TRWS) [Kolmogorov 2006] instead selects some order for edges and vertices in the graph and sequentially updates the trees in that order. This updating operation occurs one vertex or edge at a time and the result is also updated for all the trees, so agreement is easier to be achieved. However, TRWS still does not guarantee global optimal solution as it is an approximate algorithm and it is of practical consideration to stop the algorithm when the bound does not improve instead of checking tree agreement condition.

2.5 Algorithms based on Linear Programming Relaxations

Linear Programming (LP) relaxation covers a rather sizable part of discrete energy minimization algorithms mainly because a large portion of algorithms can be expressed or reinterpreted as LP of some sort. Several algorithms mentioned in earlier sections such as QPBO and TRWS can also be categorized into this class as well. LP relaxation has an advantage of providing a lower bound as the calculation progresses but the relaxation itself means that the solution is no longer constrained to be in the form of an answer to the original problem.

For binary energy minimization, LP relaxation algorithms generally relax the label set from the discrete binary set $\{0,1\}$ to the continuous set of an interval $[0,1]$.

For variables $p, q \in \mathcal{V}$, the unary and pairwise terms are inserted into the objective function as

$$\theta_p(x_p) = (1 - x_p)\theta_{p,0} + x_p\theta_{p,1} \tag{2.10}$$

and

$$\begin{aligned} \theta_{pq}(x_p, x_q) = (1-x_p)(1-x_q)\theta_{pq,00} + (1-x_p)x_q\theta_{pq,01} \\ +x_p(1-x_q)\theta_{pq,10} + x_p x_q \theta_{pq,11} \end{aligned} \tag{2.11}$$

resulting in a pseudo-Boolean energy function

$$\begin{aligned} E(\mathbf{x}) = \theta_{const} + \sum_{p\in\mathcal{V}} (\theta_{p,0}(1-x_p) + \theta_{p,1}x_p) \\ + \sum_{pq\in\mathcal{E}} (\theta_{pq,00}(1-x_p)(1-x_q) + \theta_{pq,01}(1-x_p)x_q + \theta_{pq,10}x_p(1-x_q) + \theta_{pq,11}x_p x_q). \end{aligned} \tag{2.12}$$

One can also imagine this objective function being the dot product of the concatenated parameter vector $\boldsymbol{\theta}$ indexed by the index set $\mathcal{I}$ introduced in section 2.2.1 and the associated variable vector.

For multi-label energy minimization, one cannot simply use the set $[0, k-1]$ in place of $\mathcal{L} = \{0, 1, ..., k-1\}$ since there is no precise means to construct the corresponding objective function for LP. Instead, a set of binary variable $\{x_{p(0)}, x_{p(1)}, ..., x_{p(k-1)}\}$ is defined to associate with each possible label for variable p. For pairwise potential associated with an edge $pq \in \mathcal{E}$, the set $\{x_{pq(i,j)} \,|\, (i,j) \in \mathcal{L}\times\mathcal{L}\}$ is used. This binary labeling must then satisfy the normalization constraints and the marginalization constraints

$$\begin{aligned} \sum_{i\in\mathcal{L}} x_{p(i)} = 1; \qquad & p \in \mathcal{V} \\ \sum_{(i,j)\in\mathcal{L}\times\mathcal{L}} x_{pq(i,j)} = x_{p(i)}; \quad & pq \in \mathcal{E}, p \in \mathcal{V} \end{aligned} \tag{2.13}$$

to be consistent with a multi-label solution. LP relaxation then removes the binary constraint and progress as in the binary case.

As can be speculated from the definition, the sheer number of these binary variables makes generic LP-solvers nigh-impossible to be of practical use in most cases except for small problems. Specialized solvers with dual formulation are customarily used instead of just primal. Dual Decomposition (DD) scheme [Komodakis *et al.* 2011] decomposes the graph of the random field to be optimized into several subgraphs with overlapping vertices and solves them separately but with Lagrange multipliers which are used to facilitate agreement condition. The work of [Kappes *et al.* 2012] extends DD further by updating the dual variables using the subgradient obtained from solving the subproblems.

2.6 Combinatorial Algorithms

A-star search and branch-and-bound search are popular searching strategies for problems that are impractical for brute-force search. For random field optimization, however, attempting to find global optimal solution by expanding partial solutions one-variable-at-a-time through the solution spaces is not suitable nor tractable because of the sheer number of possible paths and branches. Instead, combinatorial search for random field optimization employs integer linear programming (ILP) techniques which use convex polytopes to define candidate solution sets, making the number of candidates viable.

There are two main ILP techniques in the literature: branch-and-bound and cutting-plane. Cutting-plane techniques start by finding a solution for LP-relaxation polytope and then repeatedly add constraints violated by the solution to the polytope until an integer solution is found. The work of [Savchynskyy *et al.* 2013] proposed using combinatorial solver only where relaxed LP solver returns non-integer labels which allows certain big problems to be solved exactly. However, combinatorial sub-problems can still become too large for practical use.

On the other hand, branch-and-bound techniques start by using the solution space as the only polytope in the candidate set and repeatedly select a polytope from the candidate set and split it into several polytopes, finding a solution for each if possible and disregarding ones with lower bounds higher than the best solution so far. Branch-and-bound is often used with other searching techniques. For instance, the work of

[Bergtholdt *et al.* 2010] uses A-star search with tree-based bounding heuristic. Branch-and-bound methods, however, scale rather poorly to large problems [Kappes *et al.* 2015].

2.7 Comparison of Inference Algorithms

It has been recognized for several decades that labeling problems can be intuitively and elegantly formulated using random field framework, but it is not until the last decade that the resulting optimization problems have been shown to be not as intractable as most researchers have thought. The algorithms mentioned in the previous sections have proven to be very powerful in their respective introductions and have translated well into other applications but the general consensus remains that no single algorithm performs best for all circumstances. The comparative study in [Tappen and Freeman 2003] found that graph-cut-based approach was able to solve for solutions with lower energy when compared to message-passing-based approach for the task of stereo vision.

The research of [Boykov and Kolmogorov 2004] provides an experimental comparison of several max-flow/min-cut algorithms for applications in the computer vision field. Although it is slower than other techniques on several types of graphs the combinatorial optimization community commonly used for testing, their own proposed technique consistently outperforms other techniques for 2D grid graphs and is the most popular implementation of graph cut technique used in the computer vision community.

The comparative study in [Szeliski *et al.* 2008] evaluates and compares the solution quality and computation time of a number of energy minimization algorithms for 2D grid-graph pixel-labeling problems and suggests that graph-cut-based approaches achieve better running time for computer vision tasks compared to message passing. They have found that expansion move-making algorithm works best across their benchmarks while TRWS performs well and even on par with move-making algorithm on certain benchmarks. LBP, surprisingly, performs rather poorly on many of their benchmarks. They also note how much better present-day energy minimization algorithms are than the longstanding ICM.

The research of [Kappes, *et al.* 2015] takes a step further and provides a more modernized and wider-ranged comparative study with OpenGM implementation and benchmarks. Their results show that different algorithms are suitable for different situations, which is not surprising since most algorithms were proposed with certain specialized purpose in mind and are often not easily generalized to other problem classes. More general-purpose algorithms that can find solutions for a large class of problems are often much slower. TRWS message passing performs well for problems to which it can be applied and the relaxation is nearly tight. Move-making algorithms perform slightly worse but do not suffer the restrictions on the conditions of problems like TRWS. Combinatorial algorithms can be utilized directly for some problems and are, in some instances, faster than some state-of-the-art algorithms. LP-relaxation algorithms are usually not restricted to problem classes and often provide good solutions but are also generally slower than other classes of algorithms. The authors also suggest that primal move-making algorithms seem to work best for the cases in which others do not fare so well.

2.8 Hierarchical Schemes for Random Field Optimization

As with most computational problems, various attempts to speed up the computation time using hierarchical approaches have been investigated in the area of random field optimization. The work of [Veksler 2006] uses hierarchical approach in graph-based stereo correspondence to increase efficiency. Input stereo pair is used to construct Gaussian pyramid and the process goes to the next finer level by restricting the disparity range using the results at the previous coarser level. Their experimental results were not encouraging as the loss in accuracy was quite considerable compared to gain in efficiency.

Gaussian-pyramid was also used in [Zhang *et al.* 2010] to reduce the numbers of labels and variables to a half and a quarter, respectively. The process keeps going down the pyramid while fine-tuning over a small range, e.g., $\{-1,0,1\}$. Merging variables together, however, can lead to wrong labeling that small range fine tuning cannot correct. This is because neighboring pixels can have completely different labels at the discontinuities in the scene where one object changes to another, which is most noticeable for slender objects.

LogCut [Lempitsky, *et al.* 2007] uses binary subdivision for hierarchical partitioning of the label space. The resulting hierarchy is in the arrangement of a binary tree with the depth of the tree equal to $\log_2(|\mathcal{L}|)$. Using binary representation of the labels, the process goes from more significant bits to less significant bits and makes hard decisions along the hierarchy. To cope with the effect of these irreversible decisions, the authors of [Lempitsky, *et al.* 2007] also proposed iterated LogCut, which considers different bit codings in each iteration. They introduce a shift s to be applied to the label values before binary encoding, i.e., a label d is instead shifted to $d+s(\mathrm{mod}\,|\mathcal{L}|)$ where mod is the modulo operation. Their hierarchical energy minimization then proceeds as before and the solution from the shifted label space is then "fused" with the best-so-far using what is now called fusion move [Lempitsky, *et al.* 2010].

The work of [Kumar and Koller 2009] uses r-HST (r-hierarchically well-separated tree) metrics to define hierarchical label clustering. Their method uses α-expansion to minimize energy in a bottom-up hierarchical sequence but has a limitation in the class of energy functions that can be applied. The same concept is also explored independently in [Delong *et al.* 2012] which proposed hierarchical fusion algorithm for a class of energy functions that has "hierarchical" cost, requiring the label space to naturally form groups. Their examples of such label spaces include "car, road, sky" in "outdoors" group and "table, chair, wall" in "indoors" group for the use in object recognition. They have shown that structured label space with energy function satisfying certain conditions can be solved more efficiently via their h-fusion algorithm, which generalizes α-expansion.

Trained classifiers are used in [Conejo *et al.* 2014] to help in pruning the labels as the process undergone Gaussian pyramid iteration. To preserve sharp edges, presence of strong discontinuity (PSD) is included as one of the features and the nodes on a strong discontinuity border need to be treated separately.

The study in [Zach 2014] proposed a coarse-to-fine approach based on primal-dual min-sum belief propagation. Their dead-end elimination detects states that are unfavorable due to having extremely large unary potentials and other possible causes that render them not to be part of any optimal solution. But because of irregular state space, fast message passing cannot be utilized.

The stereo correspondence algorithm in [Taniai *et al.* 2014] uses pixel and region label sets to store candidate labels for pixels and regions to be proposed and fused via graph cuts. The process repeatedly finds new solution by drawing new proposals for each pixel from the union of its own pixel label set, its neighbors' pixel label sets, its own region label set, and its neighbors' region label sets.

The work of [Meir *et al.* 2015] proposes variable grouping using conditional entropy to gauge confidence of using label assigned to representative variable as label of other variables in its group. Their variable grouping results in a spatial hierarchy that can be optimized with inference algorithms that satisfy certain requirements.

The work of [Li *et al.* 2016] shows that there could be no approximate polynomial-time algorithm with reasonable bound on quality of the solution, even in binary pairwise energy minimization and planar case with more than two labels. More specifically, their finding precludes the existence for general energy minimization to have any approximation algorithm with a sub-exponential approximation ratio in the input size. As a corollary, this also precludes the possibility of hierarchical polynomial-time algorithm optimization algorithm to have a reasonable bound.

Chapter 3: The Proposed Methodology

While general first-order energy functions assume the form shown in Eq. (2.3), computer vision tasks often have characteristics that can be exploited which may lead to better performance. Objects are often composed of different structures at different scales and there are many problems which are often solved in a coarse-to-fine manner. The fact that the labels of certain formulated energy functions represent real-world physical quantities enables us to make use of their regularities to avoid exhaustive search of the solution spaces and, thus, gain speedup in running time while still offering comparable energy results.

Consider dense stereo correspondence as an example. In this problem, some regions may contain very little or no distinctive features and must calculate their disparity by relying on their neighbors' disparity assignments. This means that the label assignment in these regions is affected mostly by pairwise potentials while their unary potentials do not vary much. In such case, the label assignment of these variables can be calculated in a coarser manner and then refined later after the assignment of more robust variables in nearby regions is done.

The proposed methodology exploits these facts by systematically grouping labels with comparable potentials together. After that, energy minimization is done in a hierarchical manner.

3.1 Local Label Hierarchies

For each random variable $p \in \mathcal{V}$, we propose constructing a label hierarchy in the form of a tree. The root of this tree hierarchy is a group consisting of every possible label and each leaf of the tree is a singleton group situated at the same depth and containing exactly one label. At each intermediate level, each label d belongs to exactly one group and we denote the group of label d in the hth level of the hierarchy of p by $\pi_p^h(d)$. Using this notation, the 0th-level group of d which contains only itself is $\pi_p^0(d) = \{d\}$ and the set of all hth-level groups of variable p, which we denote by $\mathcal{L}_p^h$, is

$$\mathcal{L}_p^h = \bigcup_{d\in\mathcal{L}} \{\pi_p^h(d)\}. \tag{3.1}$$

Each group g can be thought of as an interval, i.e., $g=[g^-,g^+]$, where g^- and g^+ denote the lower bound and upper bound of the labels in the group and the interval is restricted to only taking on discrete values. The set of neighbors of an hth-level group g within range $r\in\mathbb{N}$ is denoted by $\nu_p^h(g,r)$ and is defined as

$$\nu_p^h(g,r)=\{f\in\mathcal{L}_p^h\setminus\{g\}:\exists d\in f\exists e\in g,|d-e|\le r\}. \tag{3.2}$$

For a group g in the $(h+1)$st-level hierarchy of p, the set of its immediate children in the hth-level is denoted by

$$\chi_p^h(g)=\{\pi_p^h(d):\pi_p^{h+1}(d)=g, d\in\mathcal{L}\}. \tag{3.3}$$

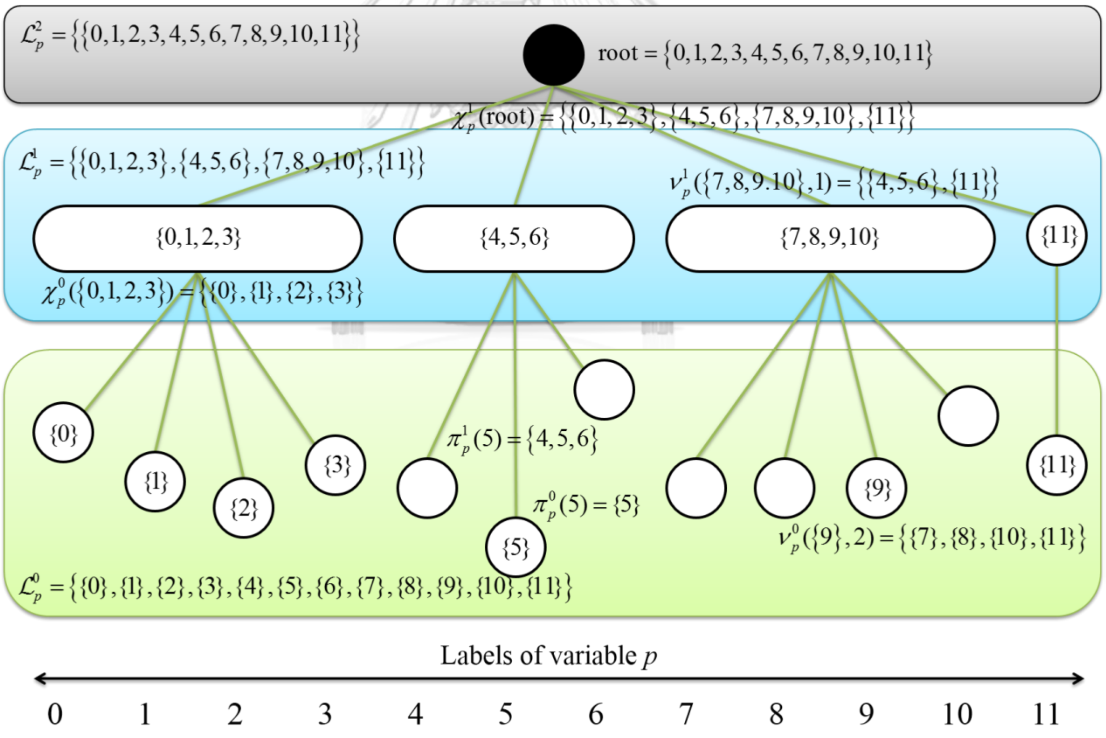

$\mathcal{L}_p^2=\{\{0,1,2,3,4,5,6,7,8,9,10,11\}\}$
$\text{root}=\{0,1,2,3,4,5,6,7,8,9,10,11\}$
$\chi_p^1(\text{root})=\{\{0,1,2,3\},\{4,5,6\},\{7,8,9,10\},\{11\}\}$
$\mathcal{L}_p^1=\{\{0,1,2,3\},\{4,5,6\},\{7,8,9,10\},\{11\}\}$
$\nu_p^1(\{7,8,9.10\},1)=\{\{4,5,6\},\{11\}\}$
$\{0,1,2,3\}$
$\{4,5,6\}$
$\{7,8,9,10\}$
$\{11\}$
$\chi_p^0(\{0,1,2,3\})=\{\{0\},\{1\},\{2\},\{3\}\}$
$\{0\}$
$\{1\}$
$\{2\}$
$\{3\}$
$\pi_p^1(5)=\{4,5,6\}$
$\pi_p^0(5)=\{5\}$
$\{5\}$
$\{9\}$
$\{11\}$
$\nu_p^0(\{9\},2)=\{\{7\},\{8\},\{10\},\{11\}\}$
$\mathcal{L}_p^0=\{\{0\},\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{10\},\{11\}\}$
Labels of variable p
0
1
2
3
4
5
6
7
8
9
10
11

Figure 9: Example of our notations in a two-level hierarchy

An example of our notations in a two-level hierarchy is depicted in Figure 9. In this case, $\pi_p^2(d)=\text{root}$ for every label d of the variable p and every group in the set of 1st-level groups $\mathcal{L}_p^1$ is a child of the root of the hierarchy; that is $\chi_p^1(\text{root})=\mathcal{L}_p^1$. It is possible for a higher-level group to have one child-group, for example, $\{11\}$, which is why level is explicitly included in the notations.

We dub this hierarchical scheme "local label hierarchy" and the grouping process "label-wise grouping." We assume, without loss of generality, that the label set $\mathcal{L}$ has a linear ordering, i.e., $\mathcal{L} = \{0,1,\ldots,k-1\}$. This is a rather common assumption in the literature [Ishikawa 2003, Kohli, *et al.* 2008, Kovtun 2003, Schlesinger and Flach 2006] and is the case of a number of applications such as image denoising and deblurring, inpainting, and stereo and multi-view reconstruction. In other applications, the order of $\mathcal{L}$ can be imposed and rearranged. The research in [Schlesinger 2007] proposes an approach to recognize if there exists an ordering of $\mathcal{L}$ in which the energy function is submodular. As pointed out by [Kohli, *et al.* 2008], a different reduction using binary indicator variables that does not depend on the order of $\mathcal{L}$ can be shown to give degenerate solutions.

3.2 Hierarchical Energy Minimization

After grouping, the problem is solved hierarchically in a top-down manner. The process starts at the root of each variable, which situates at the highest level in the configuration hierarchy and contains every possible label. In each iteration, each random variable assumes a group of labels in the next immediate lower level by minimizing an energy function. The process continues until every variable is assigned a singleton group, i.e., the 0th-level configuration is reached.

More concretely, we define an hth-level energy configuration $\mathbf{x}^h$ to be drawn from in the set

$$\mathbf{x}^h \in \prod_{p\in\mathcal{V}} \chi_p^h(x_p^{h+1})$$

where $\prod$ is the $|\mathcal{V}|$-ary Cartesian product and the $(h+1)$st-level configuration

$$\mathbf{x}^{h+1} = (x_p^{h+1} \mid p \in \mathcal{V}) \in \prod_{p\in\mathcal{V}} \mathcal{L}_p^{h+1}$$

is from the previous iteration. The h th-level energy minimization problem is to solve for a configuration $\mathbf{x}^h$ that minimizes the energy function

$$E(\mathbf{x}^h \mid \theta^h, \mathbf{x}^{h+1}) = \sum_{p \in \mathcal{V}} \theta_p^h(x_p^h) + \sum_{pq \in \mathcal{E}} \theta_{pq}^h(x_p^h, x_q^h) + \theta_{const}^h \tag{3.4}$$

where the domains of θ_p^h and θ_{pq}^h are defined to be the set $\mathcal{L}_p^h$ and $\mathcal{L}_p^h \times \mathcal{L}_q^h$, respectively. The values of the potentials are discussed later in this chapter.

Given the h th-level energy function, a binary energy function is constructed as described in section 2.2.2. The group indices start form 0 to $|\chi_p^h(x_p^{h+1})|-1$ and the set of graph vertices is defined as

$$V^h = \left\{(p,g) : p \in \mathcal{V}, g \in \{1, 2, \ldots, |\chi_p^h(x_p^{h+1})|-1\}\right\} \tag{3.5}$$

to correspond with the $(h+1)$ st-level configuration $\mathbf{x}^{h+1}$.

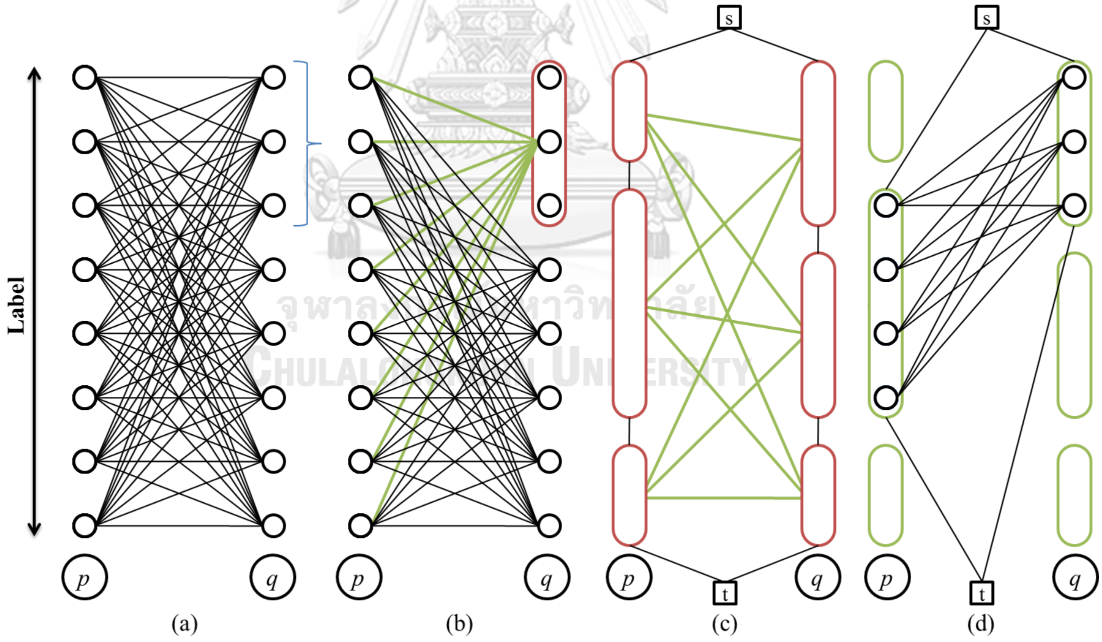


Figure 10: An example of two-level local label hierarchy in MQPBO setting

An illustration of how our local label hierarchy can reduce MQPBO graph complexity in a two-level hierarchy is shown in Figure 10. In Figure 10 (a), the three top-most labels of variable q is grouped together as shown in Figure 10 (b), which leads to the underlying graph having fewer number of edges. The 1st-level graph after grouping is shown in Figure 10 (c). The process starts at the 2nd-level, which is the root of the hierarchy. After energy minimization is done, the 1st-level configuration is obtained and the process continues to find the solution for the graph shown in Figure 10 (d) to obtain the 0th-level configuration, which is the solution.

It can be seen that this method results in a faster running time. Suppose, for instance, that the max-flow algorithm being used has time complexity of $O(|V|^3)$ and that grouping with two-level hierarchy has the average size of k^*. Using MQPBO method as the underlying inference algorithm, the running time of our approach would be $O((|\mathcal{V}|k^*)^3 + \left(|\mathcal{V}||\mathcal{L}|/k^*\right)^3)$ which is faster compared to $O(\left(|\mathcal{V}||\mathcal{L}|\right)^3)$ of the normal running time. Fewer number of variables and edges also means that the memory required for calculation is reduced. In a four-connected grid random field, the number of edges required for MQPBO would be $O((|\mathcal{V}||\mathcal{L}|)^2)$ in the normal case and $O(\max((|\mathcal{V}|k^*)^2, (|\mathcal{V}||\mathcal{L}|/k^*)^2))$ using local label hierarchy.

3.3 Group Potential Assignment

Before energy minimization can be done, the values of the group potentials in the energy function must be assigned. If each group has a representative label, one sensible choice would be to assign group potentials with the potentials of representative labels of the groups. That is, suppose the hth-level groups f and g of variable p and q have representative labels $rep(f)$ and $rep(g)$, respectively, the unary and pairwise group potentials would be assigned with

$$\theta_p^h(f) = \theta_p(rep(f))$$

and

$$\theta_{pq}^h(f,g) = \theta_{pq}(rep(f), rep(g)). \tag{3.6}$$

Aggregate functions are also natural choices for assigning group potentials. Our preliminary experiment suggests that using lower or upper bound approximations (minimum or maximum values) can often be overly-optimistic which may result in poor solutions for some cases.

We also experiment with using (uniform) mean and weighted mean of potentials of the labels in the group as group potentials. The mean for unary potentials is computed from the labels in each group as

$$\theta_p^h(f) = \frac{\sum_{d\in f} w_d \theta_p(d)}{\sum_{d\in f} w_d} \tag{3.7}$$

and the mean for pairwise potentials is computed from all possible pairs of labels in the two participating groups of the participating neighboring variables as

$$\theta_{pq}^h(f,g) = \frac{\sum_{d\in f, e\in g} w_{de} \theta_{pq}(d,e)}{\sum_{d\in f, e\in g} w_{de}}. \tag{3.8}$$

For uniform mean, the weights are set to

$$w_d = \frac{1}{|f|}$$

and

$$w_{de} = \frac{1}{|f||g|} \tag{3.9}$$

and, for weighted mean,

$$w_d = \exp\left(-\frac{(d - rep(f))^2}{2\sigma_f^2}\right)$$

and

$$w_{de} = \exp\left(-\frac{(d - rep(f))^2}{2\sigma_f^2} - \frac{(e - rep(g))^2}{2\sigma_g^2}\right). \tag{3.10}$$

Since θ_{const} is not affected by grouping, $\theta_{const}^h = \theta_{const}$ for every level h in the hierarchy. Also, we use pre-computed cumulative sum in our experiment to compute the uniform mean of each group in constant time.

3.4 Label-wise Grouping Techniques

In this section, we present three heuristics for label-wise grouping: local minimum search, cluster analysis, and maximum difference subdivision.

3.4.1 Local Minimum Search

As its name suggests, local minimum search groups labels by using local minima. That is, each local minimum, or "basin", is assigned its own group and other labels "drift" toward where their basins are. These local minima are used as representative label of the groups.

More concretely, our local minimum search technique groups labels in a bottom-up manner starting at $\mathcal{L}_p^0$, the set of all 0 th-level groups of variable p. Given $\mathcal{L}_p^h$, the set of h th-level local minima is defined as

$$\begin{aligned} & LM_p^h = \{g \in \mathcal{L}_p^h : \forall f \in \nu_p^h(g), \theta_p^h(g) < \theta_p^h(f)\} \\ & \cup \{g \in \mathcal{L}_p^h : \exists r > 0, \left(\forall f \in \nu_p^h(g,r), \theta_p^h(g) = \theta_p^h(f)\right) \\ & \wedge \left(\exists f \in \nu_p^h(g,r+1), \theta_p^h(g) \neq \theta_p^h(f)\right)\} \end{aligned} \tag{3.11}$$

where $\nu_p^h(g)$ is used to denote $\nu_p^h(g,1)$, the adjacent neighbors of a group g. From the definition, each label compares its unary potential with the potentials of its neighbors and identifies itself as a local minimum if its potential is less than all those of its neighbors or it is at the center of an equi-potential basin.

The basin $\beta_p^h(g)$ toward which a group g drifts is then defined recursively as

$$\beta_p^h(g) = \begin{cases} g & \text{if } g \in LM_p^h \\ \beta_p^h(f) & f \in \nu_p^h(g); \forall e \in \nu_p^h(g), \theta_p^h(f) < \theta_p^h(e). \end{cases} \tag{3.12}$$

That is, a label stays to be a representative label if it is a local minimum. Otherwise, it is assigned to the group of the neighbor that has the least potential among all of its neighbors. If the neighbor is not assigned a group, the search for local minimum continues until one is found. In case of a tie between the potentials of the neighbors, we choose to move the label toward its nearest local minimum. Also, we currently choose the center of the "plateau" (an equi-potential basin) as the local minimum if there are three or more consecutive labels with equal unary potentials.

The set of $(h+1)$ st-level groups $\mathcal{L}_p^{h+1}$ is the union of the groups of all local minima, that is

$$\mathcal{L}_p^{h+1} = \bigcup_{g \in LM_p^h} \left\{ g \cup \{ f \in \mathcal{L}_p^h : \beta_p^h(f) = g \} \right\}. \tag{3.13}$$

If larger groups are preferred, value greater than 1 can be used in $v_p^h(g)$. Conversely, one can also limit how large the groups can be to favor smaller groups, i.e., each basin grows to collect their neighbors until a predetermined maximum group size is reached, and the search for new basins and the group assignment process continue if there are still h th-level groups left.

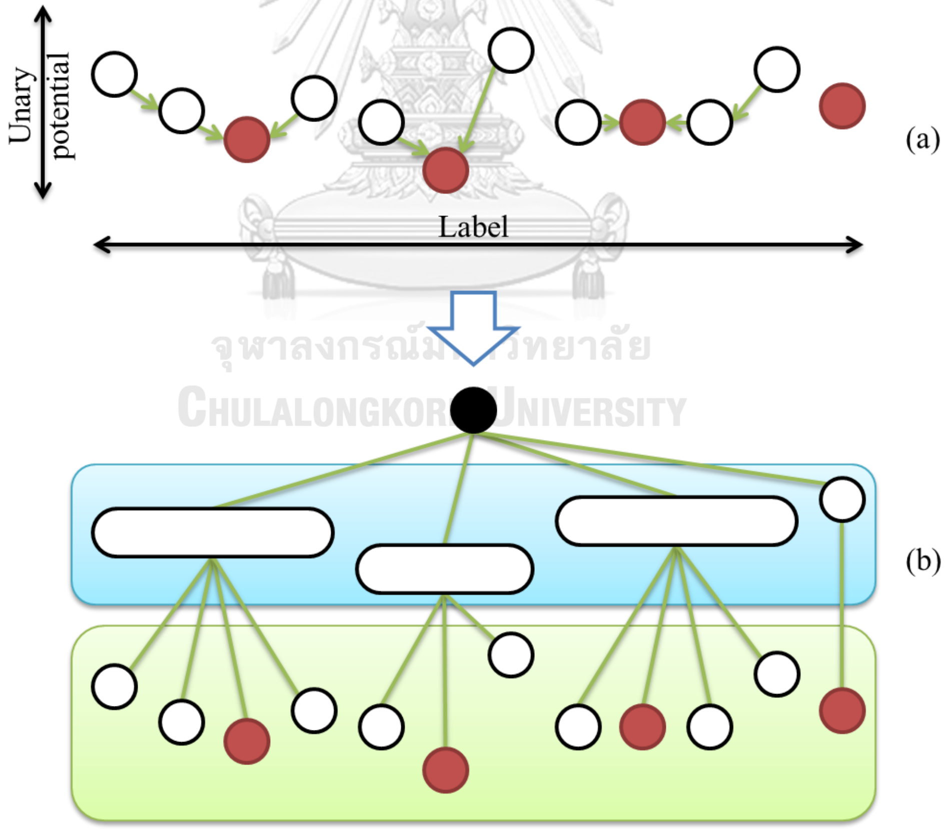


Figure 11: Label-wise grouping using local minimum search

Because one group in every pair of neighboring groups must have its unary potential less than or equal to that of the other, local minimum search grouping technique decreases the number of groups as we go up each level by at least half. An example of using local minimum search to construct a two-level hierarchy is shown in Figure 11. The arrows in Figure 11 (a) show where the labels drift toward. The resulted hierarchy is shown in Figure 11 (b). The node at the top is the $root$. The lower box contains the leaves and the upper box contains their 1st-level groups.

3.4.2 Cluster Analysis

Our cluster analysis grouping technique uses similarity in unary potentials as a basis for grouping. Here, we use a mode seeking algorithm based on mean shift [Cheng 1995, Comaniciu and Meer 2002] but other algorithms of this sort would fit our framework as well.

Given an hth-level group $g \in \mathcal{L}_p^h$ of a random variable p, the weighted unary potential mean at g within neighborhood range r is defined as

$$m_p^h(g,r) = \frac{\sum_{f \in v_p^h(g,r)} K(\theta_p^h(f) - \theta_p^h(g)) rep(f)}{\sum_{f \in v_p^h(g,r)} K(\theta_p^h(f) - \theta_p^h(g))} \tag{3.14}$$

where $K(\cdot)$ is the Gaussian kernel $K(\cdot) = \lambda \exp\left(-(\cdot)^2 / 2\sigma^2\right)$. One can imagine $m_p^h(g,r)$ as a measure that indicates which direction the group g is more similar to its neighbors in terms of potentials as the kernel $K(\cdot)$ gives more weight when $\theta_p^h(f)$ is closer to $\theta_p^h(g)$.

We call a group g a mean-shift cluster center if $m_p^h(g,r)$ is no further from $rep(g)$ than a predetermined threshold ϵ. The set of all h-level cluster centers is then defined as

$$MC_p^h = \left\{ g \in \mathcal{L}_p^h : \left| m_p^h(g,r) - rep(g) \right| < \epsilon \right\}. \tag{3.15}$$

Like local minimum search, we give each cluster center its own group. Other groups then use their $m_p^h(g,r)$ to determine which direction it should be grouped with its neighbors. The cluster center $\kappa_p^h(g)$ toward which each group g converges is defined recursively as

$$\kappa_p^h(g)=\begin{cases} g & \text{if } g\in MC_p^h \\ \kappa_p^h(f) & f\in \nu_p^h(g,1);\operatorname{sgn}(rep(f)-rep(g))=\operatorname{sgn}(m_p^h(g,r)-rep(g)) \end{cases} \tag{3.16}$$

where $\operatorname{sgn}(\cdot)$ is the signum function. The set of $(h+1)$ st-level groups is then the union of the groups of all cluster centers and is defined as

$$\mathcal{L}_p^{h+1}=\bigcup_{g\in MC_p^h}\left\{g\cup\{f\in\mathcal{L}_p^h:\kappa_p^h(f)=g\}\right\}. \tag{3.17}$$

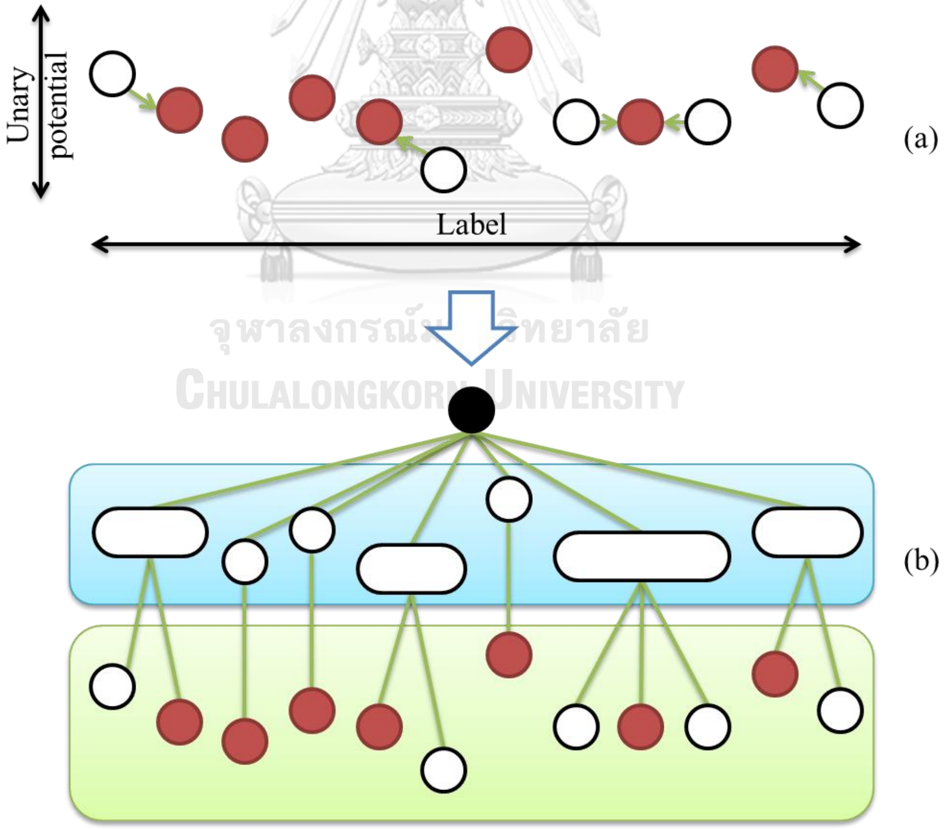


Figure 12: Label-wise grouping using cluster analysis

An example of constructing a two-level hierarchy using mean shift cluster analysis is shown in Figure 12. Mean shift tries to move label toward the direction that has similar potentials as shown by the arrows. Compared to Figure 11, the groups resulted from using local minimum search can differ more in terms of potentials whereas the labels within each group of cluster analysis have more closely-related potentials. Another noteworthy behavior of using mean shift for grouping is that labels that situate in a neighborhood with comparable potentials would be assigned its own group and, thus, would be treated with more attention near the root of the hierarchy.

3.4.3 Maximum-Difference Subdivision

Unlike previously mentioned grouping techniques which work from the bottom upward, maximum-difference subdivision builds groups in a top-down manner. This proposed grouping technique starts at the root group and repeatedly splits each group at the points between pairs of adjacent labels with maximum unary potential differences. Because of the dividing method, there is no obvious choice for assigning representative label for each child group like previous grouping techniques.

The resulting number of subgroups can be fixed, predetermined or conditioned on certain criteria. Using fixed number of groups, such as always splitting into 2 or 4 subgroups, is straightforward but do not take the label space size into account. In contrast, predetermined number of groups considers the number of possible labels and other relevant information. For example, one can use the (desired number of levels)th root of the label space size as the number of subgroups to be split if given a desired number of levels. The number of the resulting subgroups can also be set independently as the (remaining number of levels)th root of the size of the group being split.

Certain strategy such as keep splitting until the remaining unary potential differences are less than a certain threshold can also be used, but the threshold must be different for each level which introduces a new task of determining suitable thresholds. This can be done by using certain criteria, such as splitting at the differences that are greater than the mean of all differences or ranking the potential differences and then splitting the top ρ percentile for some parameter ρ. These criteria loosely resemble using predetermined number of groups but are more complicated.

Let $diff_p(a) = \left|\theta_p(a) - \theta_p(a+1)\right|$ be the potential difference between a pair of adjacent labels a and $a+1$ of variable p with $0 \le a < k-1$. To obtain n groups, the first $(n-1)$ maximum potential differences are selected.

Given a group $g \in \mathcal{L}_p^h$ at level h, the set of maximum-potential-difference labels of g is defined as

$$MD_p^h(g) = \left\{a \in g : \forall b \in g \setminus MD_p^h(g), diff_p(a) \ge diff_p(b)\right\} \tag{3.18}$$

with the constraint $|MD_p^h(g)| = \min(n-1, |g|-1)$. The set of its immediate children $\chi_p^{h-1}(g)$ is constructed as described in Table 2.

Table 2: Pseudocode for maximum-difference subdivision

```
Input:  $g \in \mathcal{L}_p^h$, a group of variable $p$ at level $h$
        $n$, the desired number of child groups
Output: $\chi_p^{h-1}(g)$, a set of child groups of $g$ at level $h-1$

Set $n_g := \min(n-1, |g|-1)$.
Select the first $n_g$ maximum-difference labels to create $MD_p^h(g)$.
Set $\chi := \{g\}$.
While $\exists f = [f^-, f^+] \in \chi \quad \exists a \in MD_p^h(g), a \in f \wedge a+1 \in f$ :
   Set $f_1 := [f^-, a]$ and $f_2 := [a+1, f^+]$.
   Remove $f$ from $\chi$ and add $f_1$, $f_2$ to $\chi$.
Endwhile

Return $\chi$.
```

The set of all $(h-1)$st-level groups is then the union of the subgroups split from all the h-level groups and is defined as

$$\mathcal{L}_p^{h-1} = \bigcup_{g \in \mathcal{L}_p^h} \chi_p^{h-1}(g) \tag{3.19}$$

In cases where there are more than one adjacent pairs of labels with equal potential differences, we currently choose to split the largest group at the point nearest to its middle. For a variable with every label having equal unary potential, this strategy would be equivalent to binary subdivision.

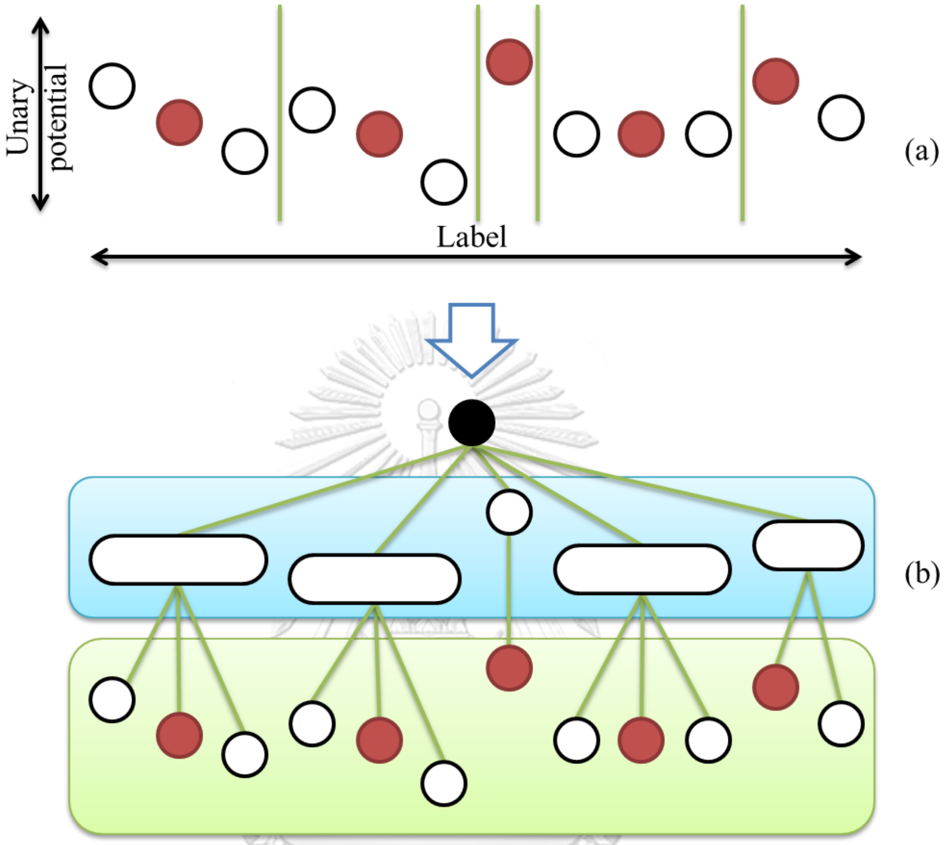


Figure 13: Label-wise grouping using maximum difference subdivision

Figure 13 shows an example of using maximum-difference subdivision in building a two-level hierarchy. Compared to Figure 12, it can be seen that the resulting groups of maximum-difference subdivision resemble those of cluster analysis to some degree. This behavior is expected as our cluster analysis technique tries to group labels with close potentials together whereas maximum-difference subdivision tries to break labels with diverse potentials apart. The advantage (or one can also view it as disadvantage) of maximum-difference subdivision is that the number of resulting groups is preset, which allows better control but also comes at a caveat. If Figure 13 had been subdivided into four groups, the left-most six labels would have been put into a single group, which would put more workload into the bottom of the hierarchy as opposed to Figure 12 that deals with these labels mostly at the top.

Further constraints can be enforced to make the number of levels logarithmic to the size of the label space. A simple case is where each group is constrained to split into only two subgroups. In this case, we can constrain the size of the subgroups not to be larger than some fraction of the group being split to limit the number of levels. For example, limiting the subgroup size to three quarters of the group size guarantees that the number of resulting levels will not go above the logarithm of the label space size to the base $^{4}\!/_{3}$. We dub this binary subdivision scheme "Skewed Log Subdivision" which can be viewed as a generalization of algorithms with logarithmic complexity proposed in [Lempitsky, *et al.* 2007] where binary subdivision does not have to occur at the middle of the group.

3.5 Discussions

Our methodology can be considered as a generalization of the bit-level subdivision proposed in [Lempitsky, *et al.* 2007] where our groups are allowed to have more than two subgroups each and also allowed to have different sizes. Similar to [Lempitsky, *et al.* 2007], our methodology makes irreversible pruning of the label space starting at the top of the hierarchy. However, binary subdivision builds very coarse and deep hierarchy, which is not fitting for applications in general. To make LogCut more robust, iterated LogCut was proposed in which the label set is shifted by some amount before bit-level subdivision in each iteration and the solutions from different shifted amounts are combined using fusion move. Our methodology attempts to integrate the information from the energy function at hand into the process of constructing the hierarchy by taking the potentials of the labels into account.

Using hierarchical scheme in stereo correspondence has provided quite discouraging quality in the results in some cases. The Gaussian pyramid employed in [Veksler 2006] did not take into account the energy function during hierarchy construction, which is the key aspect of our methodology. Also, using pyramid scheme can result in over-smoothed edges because the higher-level variable does not allow pixels in its group to have much diverse disparity values that occur at edges [Zhang, *et al.* 2010].

Our idea is related to [Kumar and Koller 2009] but their hierarchical graph cuts approach expects the energy function to be r-HST metric. The concept of [Delong, *et al.* 2012] is related to ours as well but, instead of having the label space placed in a linear ordering which is the case of our work, requires structured label space that is explicitly grouped in hierarchy.

Using machine learning for coarse-to-fine pruning of label space has been proposed in [Conejo, *et al.* 2014]. Learning, however, requires at least one training set from which features are drawn, as opposed to our work which relies exclusively on information from the given energy function and therefore does not require training.

Unlike other works, our local label hierarchy scheme builds, for each variable, its own label hierarchy by drawing information from the potentials instead of forcing every variable to use the same hierarchy. We also do not employ spatial grouping of random variables which has been shown in several works to oftentimes inadequately preserve discontinuities. The additional processing steps of our methodology have theoretical complexity significantly less than that of the overall optimization process. The empirical performance of our methodology will be demonstrated and discussed in the next chapter.

Chapter 4: Experiments and Results

We have applied our methodology to computer vision applications with linearly ordered label space, namely, dense stereo correspondence, image denoising, and image inpainting. We have also tested our methodology with a benchmark database of discrete energy minimization problems [Kappes, *et al.* 2015] which offers a mixed range of diverse types of models. Performance of our methodology has been compared with our implementation of MQPBO method [Kohli, *et al.* 2008], α-expansion algorithm [Boykov, *et al.* 2001], and algorithms with logarithmic complexity introduced in [Lempitsky, *et al.* 2007].

4.1 Algorithm Implementation

For α-expansion algorithm, each sweep iterates through all possible labels through a series of binary graph cuts moves. Here, the α-expansion algorithm has been modified to be QPBO-based so that it can cope with non-submodular energy and the comparison can be equitably made. For cases in which QPBO method returns partial solutions, we break the tie by assigning each unlabeled pixel the label with lower unary potential between the two competing ones. The initial configurations of α-expansion are different in each of the benchmark application being tested and will be provided shortly in their related section.

Algorithms with logarithmic complexity [Lempitsky, *et al.* 2007] build their hierarchies by partitioning the label space by the bit values of its binary encoding. The final labeling of each pixel is solved hierarchically via fusion move binary optimization steps [Lempitsky, *et al.* 2010]. The simple binary subdivision (**Log Simple**), minimum (**Log Min**), and mean (**Log Mean**) variances are compared in the experiment. These variances differ in their methods of group potential assignment. Minimum value for each group is taken from every possible value of the less significant bits. Since all undetermined bits are essentially of equal importance, mean values are computed without weight. Simple binary subdivision assigns potential to each group by setting the less significant bits to zero.

The techniques proposed in this work being compared are **LocalMin Min**, **LocalMin Mean**, **Cluster Center**, **Cluster Mean**, **MaxDiff Mid**, **MaxDiff Mean**, **SkewLog 2/3**, and **SkewLog 3/4**. Algorithms followed by **Mean** use mean value calculated from all labels in a group as group potential. **Min** uses minimum unary potential as group potential for **LocalMin** (section 3.4.1), **Center** uses potential of the cluster center as group potential for **Cluster** (section 3.4.2), and **Mid** uses the potential of the middle label of the group as group potential for **MaxDiff** (section 3.4.3) since it does not have a clear choice for representative label. The numbers specified after **MaxDiff**, such as **8** and **16**, indicate the predetermined numbers of subgroups. **SkewLog** is a special case of **MaxDiff** where each group is split into two subgroups and the fractions **2/3** and **3/4** indicate the constrained sizes of the subgroups are not to be larger than two thirds and three quarters of the size of the group being split, respectively. For a label space size of 256, **Log** would yield $\lceil \log_2(256) \rceil = 8$ hierarchical levels whereas **SkewLog 2/3** and **3/4** would yield no greater than $\left\lceil \log_{3/2}(256) \right\rceil = 14$ levels and $\left\lceil \log_{4/3}(256) \right\rceil = 20$ levels, respectively.

In our experiment, we have used the QBPO implementation from [Rother *et al.* 2007] which uses the max-flow algorithm from [Boykov and Kolmogorov 2004]. Despite not having any computational bound, this max-flow algorithm has been shown to outperform many methods with polynomial-time bound for optimizing several two- and three-dimensional grid graphs in computer vision and is the most widely-used implementation of graph cut technique in computer vision research.

4.2 Stereo Benchmarks

Stereo correspondence has been one of the most intense areas of research in computer vision for decades. The problem takes two images from different viewpoints of a scene as inputs and asks to match each pair of locations in the images that correspond to the same physical scene point. In contrast with sparse stereo correspondence where only a subset of image locations with visually distinctive features are matched, dense stereo correspondence outputs a disparity map of the whole scene (every pixel has an assigned disparity) and, thus, is more suitable for further use in higher-level vision applications, such as object tracking and classification, and reconstructing the 3D scene structure.

We formulate dense stereo correspondence as a multi-label energy minimization problem in which each pixel is considered a random variable and the matching cost and smoothness prior constitute unary and pairwise potentials of the corresponding energy function, respectively. The matching cost is used in stereo problems to evaluate the similarity between image locations whereas the smoothness prior penalizes difference in disparity of neighboring pixels and help coping with ambiguous regions in the image pair such as those caused by occlusions or lack of distinctive features.

In our experiment, we use sum of squared differences and sum of absolute differences as matching costs. These differences are computed from primary (left) and secondary (right) RGB images and multiplied with Gaussian weights before summing over circular neighborhood window using with the sampling-insensitive computation in [Birchfield and Tomasi 1998]. A penalty term is used when a pixel in the primary image is matched to the imaginary pixel outside the range of the secondary image to reduce the effect near the edge.

We switch between different smoothness priors to experiment with global smoothness constraints whose purposes are to penalize the disparity differences between neighboring pixels. Our selection of functions consists of constant function, linear function, inverse function, and Gaussian function. These functions are calculated based on the intensity gradient in the primary image to reflect the fact that the more different two neighboring pixels appear, the more they are likely to have different disparities and thus should not be penalized as much as the case where two neighboring pixels appear to be similar.

Suppose neighboring pixels p and q were to have disparity values i and j, respectively, the constant function is calculated based on only the difference in their disparity values, i.e., $\theta_{pq}^{const}(i, j) = \lambda_{const}\left|i - j\right|$. On the other hand, linear function, inverse function and Gaussian function are calculated based on both the intensity gradient and the disparity difference and are defined as

$$\theta_{pq}^{lin}(i, j) = \lambda_{lin}\left|i - j\right|(max\_intensity - \left|I(p) - I(q)\right|),$$

$$\theta_{pq}^{inv}(i, j) = \frac{\lambda_{inv}\left|i - j\right|}{\left|I(p) - I(q)\right|}$$

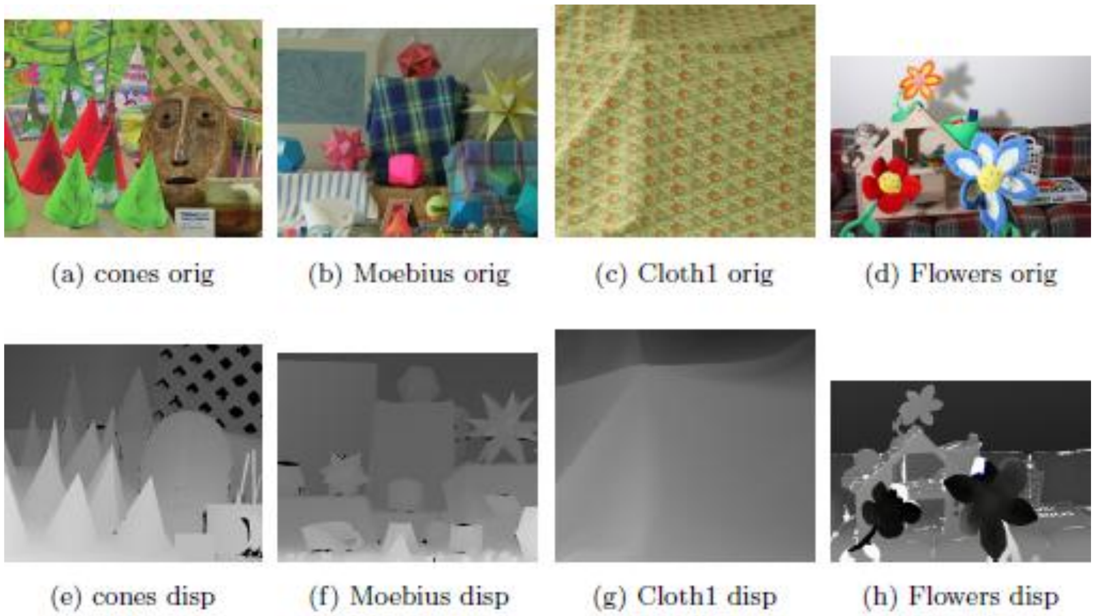
and
$$\theta_{pq}^{Gauss}(i, j) = \lambda_{Gauss}\left|i - j\right| e^{-\frac{(I(p)-I(q))^2}{h}} \qquad (4.1)$$

where $I(p)$ and $I(q)$ are the intensity values of pixel p and q in the primary image, $max\_intensity$ is the maximum intensity value allowed in the images, λ and h are parameters to be set in the experiment settings.

In each setting, the parameter combination used is the one that, when solved using MQPBO method, has given lowest error result when compared with ground truth images. All terms in the energy function were computed prior to energy minimization and then used throughout the experiment. The initial labeling for each pixel of α-expansion algorithm is chosen to be assigned with the disparity that has the lowest unary potential in that pixel.

The stereo datasets we used are from the Middlebury Stereo Vision[13] [Hirschmuller and Scharstein 2007, Scharstein *et al.* 2014, Scharstein and Pal 2007, Scharstein and Szeliski 2002, Scharstein and Szeliski 2003] which have been growing and improving over more than a decade and have been used as the de facto benchmark for stereo correspondence and multi-view reconstruction for almost as long. The datasets provide rectified stereo image pairs with radial distortion removed and pixel-accurate ground-truth disparity maps. Some examples of the ground truth disparity maps and the corresponding original primary (left) images from the datasets are shown in Figure 14.

[13] URL: vision.middlebury.edu/stereo/

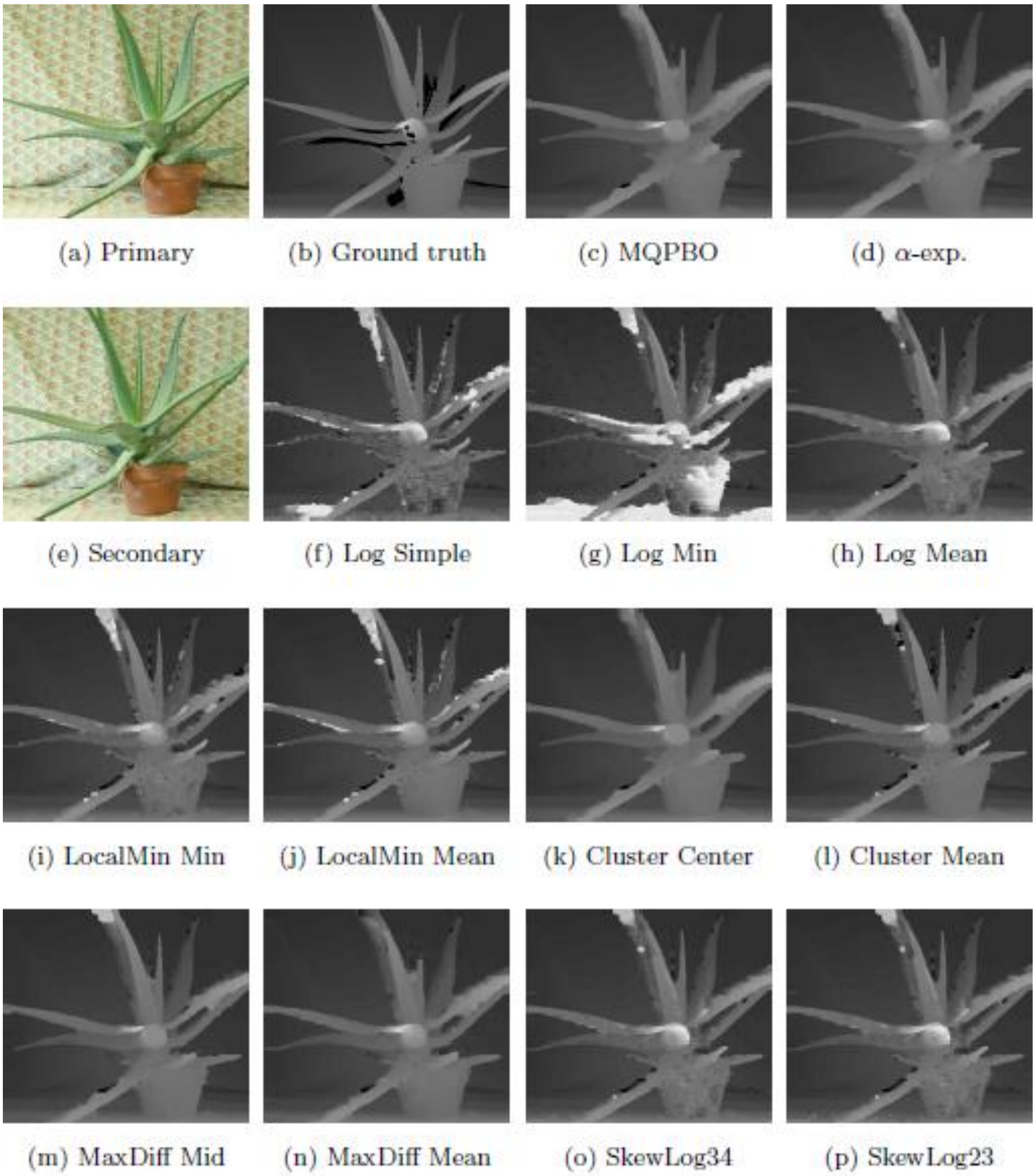


(a) cones orig (b) Moebius orig (c) Cloth1 orig (d) Flowers orig

(e) cones disp (f) Moebius disp (g) Cloth1 disp (h) Flowers disp

Figure 14: Examples from the stereo datasets (From left to right: taken from the 2003, 2005, 2006, and 2014 datasets.)

The input images and ground truth image from the Aloe stereo pair are shown in Figure 15, as well as the disparity maps of the solutions computed by the techniques being compared using sum of square differences and Gaussian smoothness prior. Qualitatively, the artifacts in **Log** techniques, especially in **Log Min**, can be seen to be quite strong. This is due to the erroneous assignment in the more significant bits, which cannot recover because the techniques employ a top-down computation scheme. Both **SkewLog** techniques also demonstrate this behavior and, upon closer inspection, **SkewLog34** appears to have smoother disparity map, which suggests that having more choices to subdivide leads to better qualitative result.



(a) Primary (b) Ground truth (c) MQPBO (d) α-exp.

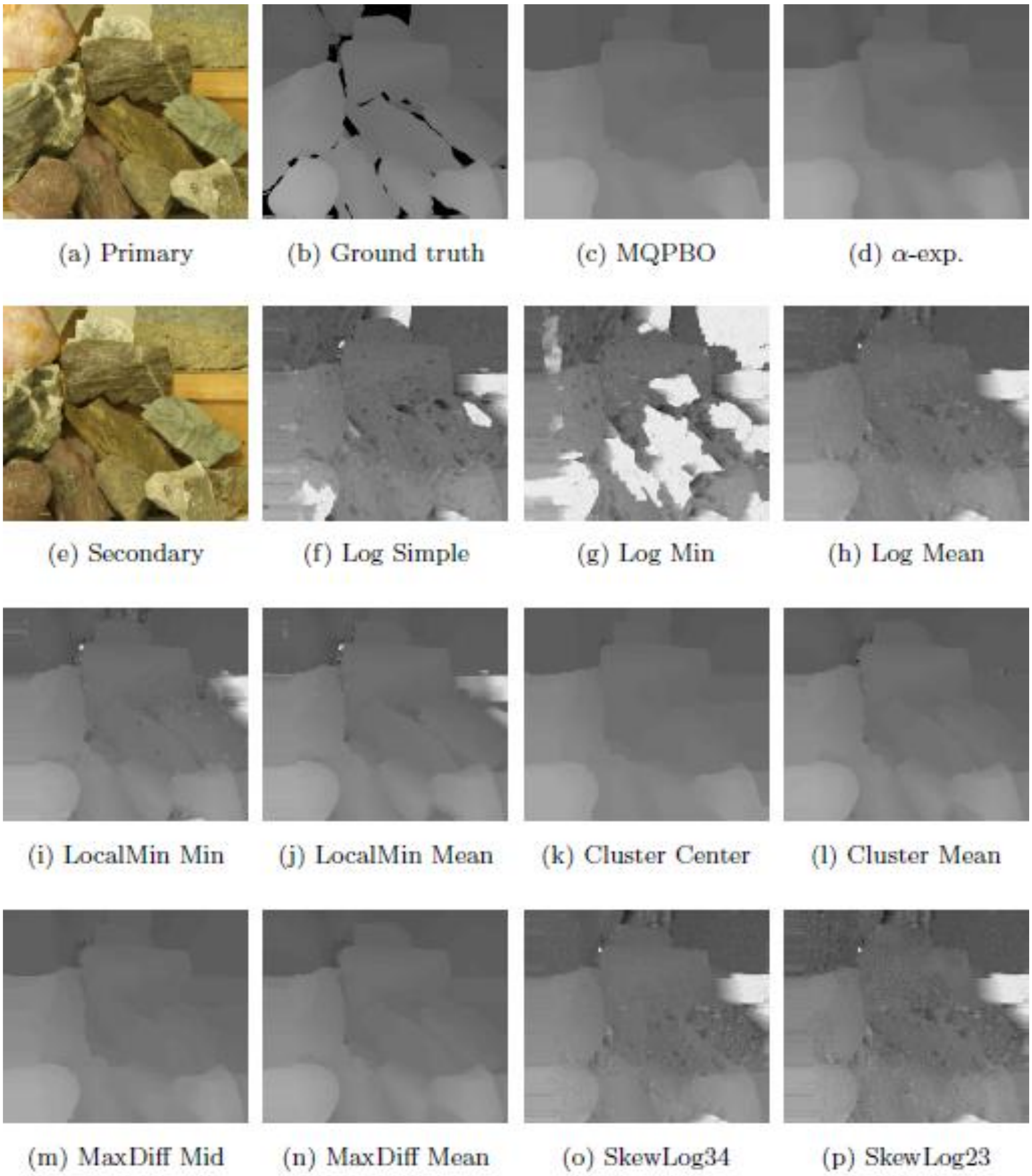
(e) Secondary (f) Log Simple (g) Log Min (h) Log Mean

(i) LocalMin Min (j) LocalMin Mean (k) Cluster Center (l) Cluster Mean

(m) MaxDiff Mid (n) MaxDiff Mean (o) SkewLog34 (p) SkewLog23

Figure 15: Disparity maps of the Aloe stereo pair

The artifacts in **LocalMin** techniques can be seen to be more pronounced than those in **Cluster** and **MaxDiff** as a result of the group potential estimation being over-optimistic. Between the two **LocalMin** techniques, extra computation of mean values in **LocalMin Mean** provided a disparity map with fewer artifacts. This behavior can also be seen in **MaxDiff**. Despite more computation steps, the disparity map from **Cluster Mean** can be seen to have more artifacts than that from **Cluster Center**. This hints that the cluster centers from mean-shift are more suitable for group potential assignment. **MaxDiff**, on the other hand, gives better result when using mean values than simply using the middle label in each group as representative label.


(a) Primary (b) Ground truth (c) MQPBO (d) α-exp.

(e) Secondary (f) Log Simple (g) Log Min (h) Log Mean

(i) LocalMin Min (j) LocalMin Mean (k) Cluster Center (l) Cluster Mean

(m) MaxDiff Mid (n) MaxDiff Mean (o) SkewLog34 (p) SkewLog23

Figure 16: Disparity maps of the Rocks2 stereo pair

The techniques being compared exhibit the same behavior in other stereo pairs as well, for example, results from the Rocks2 stereo pair as can be seen in Figure 16. It is worth pointing out that the results from this stereo pair have regions where the disparity values of the background (the surface that the rocks were placed on) are not zero in the results but are zero in the ground truth, which is expected since the background appears to lack any distinctive features so the disparity values were inferred from the rocks through pairwise potentials.

Table 3: Speedup, energy ratio and error results from stereo experiment

Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	Error	Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	Error
MQPBO	1	---	18.33%	MaxDiff8 Mid	8.47	+9.20%	18.51%
AlphaExp	1.1	+0.002%	18.29%	MaxDiff8 Mean	8.66	+5.65%	18.43%
Log Simple	17.1	+86.29%	27.85%	MaxDiff16 Mid	9.42	+5.52%	18.61%
Log Min	14.6	+104.83%	28.25%	MaxDiff16 Mean	9.7	+4.23%	18.32%
Log Mean	16.2	+71.20%	27.81%	MaxDiff24 Mid	9.03	+5.04%	18.41%
LocalMin Min	9.16	+30.98%	20.24%	MaxDiff24 Mean	9.28	+4.41%	18.34%
LocalMin Mean	9.23	+17.79%	20.07%	MaxDiff32 Mid	8.49	+6.56%	18.39%
Cluster Center	7.99	+4.42%	18.36%	MaxDiff32 Mean	8.65	+6.31%	18.37%
Cluster Mean	7.16	+9.77%	18.52%	MaxDiff48 Mid	7.31	+9.80%	18.52%
SkewLog 3/4	12.9	+62.39%	25.27%	MaxDiff48 Mean	7.64	+9.78%	18.52%

Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	Error	Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	Error
SkewLog 2/3	13.9	+64.02%	26.31%				

Table 3 summarizes speedup, energy ratio, and error results from all settings of stereo experiment over all stereo pairs. Compared to the baseline techniques (**MQPBO** and **AlphaExp**), **Log** techniques give the best speedup but also the worst energy and error. **SkewLog** techniques give better energy and error rate than **Log** with less speedup, which supports our thesis that incorporating unary potentials into the subdivision process can lead to higher-quality results in terms of energy and error rate. **LocalMin** techniques are the next fastest and give better energy and error rate. Notice that both **Log** and **LocalMin** perform worse when using **Min** than **Mean**, suggesting that lower bound is not a good candidate for group potential assignment.

As we reported in [Leelhapantu and Chalidabhongse 2018], **Cluster Center**, the most competitive technique at the time, provides a near order-of-magnitude speedup with less than 5% increase in energy. **Cluster Mean**, however, gives less speedup and larger increase in energy, hinting that using mean-shift center is better for group potential assignment than using mean value.

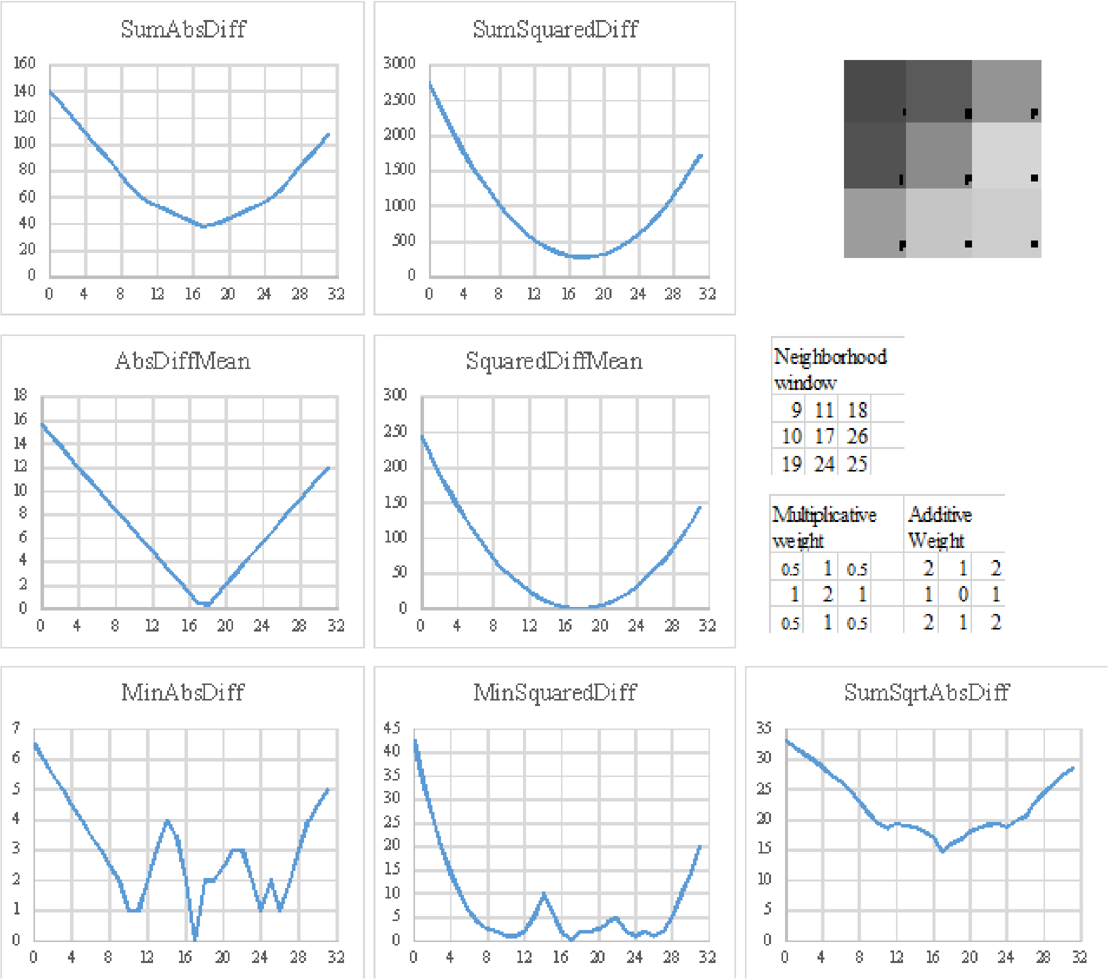
We experiment with several numbers of subgroups in **MaxDiff** techniques. The results demonstrate that **MaxDiff16** performs best on all counts: speedup, energy, and error with respect to ground truth. While we have expected using 16 as number of subgroups to give the best speedup, we did not expect the energy and error results to be this competitive. Taking a closer inspection has revealed that while using larger number of smaller subgroups intuitively should have given the best energy and error, the reality is that having several small groups at the higher hierarchy level can lead to the corresponding configuration getting stuck at a local minimum with no room to wiggle and refine at the lower level. Using **MaxDiff8**, nevertheless, performs worse and the optimal trade-off spot appears to be at 16 where the group sizes are balanced between the hierarchy levels. Also, the difference between using **Mid** and **Mean** as group potentials are more pronounced as the number of subgroups gets smaller (the group sizes get larger).

4.3 Image Denoising Benchmarks

Digital cameras capture images by measuring light intensity reflecting from objects in the scenes. Even with constant light source, the number of photons received by each pixel in a camera can fluctuate. Furthermore, heat spurious photons can also occur if the capturing element is not adequately cooled. The resulting perturbation is called noise and the problem of estimating an underlying function from error-contaminated observations is called denoising.

Like dense stereo correspondence, we cast image denoising as multi-label energy minimization problem. The unary potential for each pixel variable measures the similarity between the new (estimated) intensity value and the observed intensity value. To help mitigate possible error in the observed value, neighboring intensity values are also included in unary potential. From preliminary experiment, using either absolute or squared difference between new intensity value and weighted mean intensity of neighborhood window for unary costs tends to cause the result to be over-smoothed because each pixel has one "preferred" value from which the unary potential is measured. This same behavior also exhibits when sum of (weighted) squared differences and sum of (weighted) absolute differences are used because both functions are convex and continue to be convex after summation which means that each of them has a global minimum and, because they are combined linearly, has progressively steeper slope as we go further from the global minimum.

To mitigate this effect, we also use minimum value of differences computed in the neighborhood window as unary cost. This way, each pixel is allowed to take on intensity values in the neighborhood window without being penalized as much as previously mentioned unary costs, which results in more edges and lines being preserved. We experiment with minimum of absolute differences and minimum of squared differences. In addition to multiplicative weight, we also insert additive weight into the differences before computing minimum, i.e., the difference value in the window center (the pixel's own observed intensity value) has 0 additive weight and each of the other difference values are added by more weight the further it is from the center. This results in unary cost with many local minima with the global minimum occurring at the pixel's own observed intensity value.



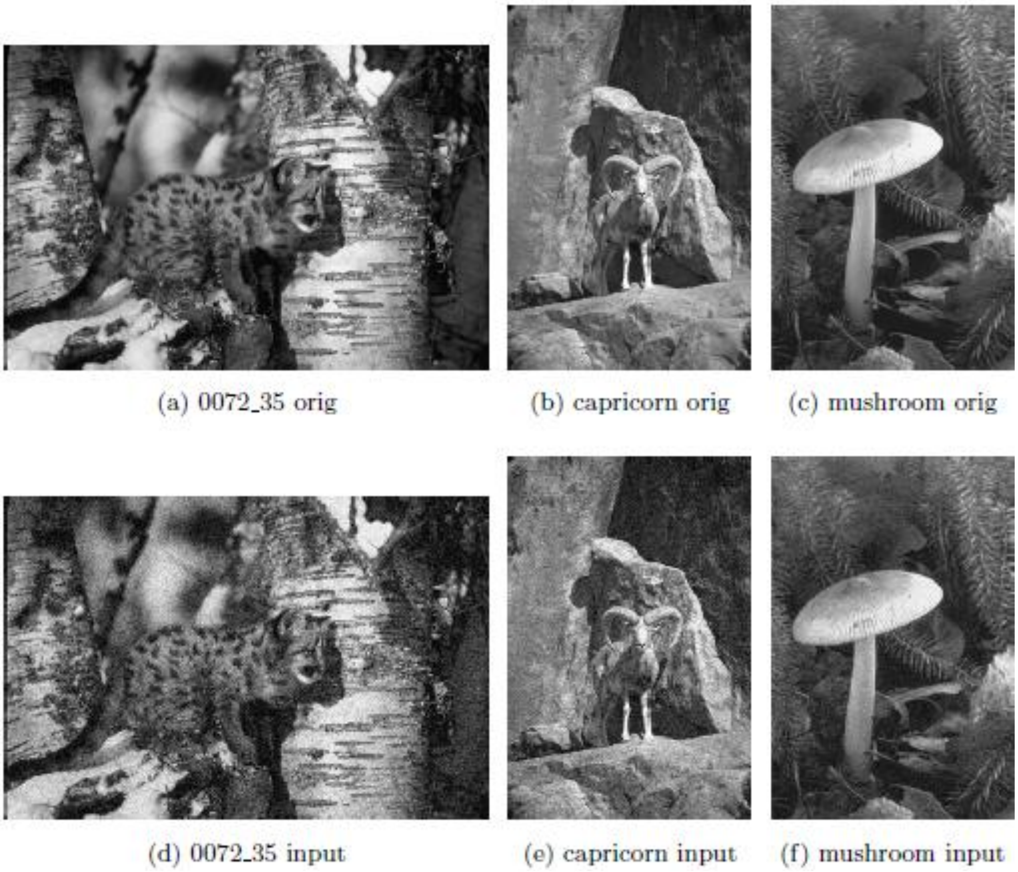
Neighborhood window		
9	11	18
10	17	26
19	24	25

Multiplicative weight			Additive Weight		
0.5	1	0.5	2	1	2
1	2	1	1	0	1
0.5	1	0.5	2	1	2

Figure 17: Behaviors of Unary Cost Functions on a sample neighborhood window

We also experiment with using sum of (weighted) square root of absolute differences as unary cost. This function by itself has a global minimum but is not convex and, thus, has multiple local minima when linearly combined together. Typical behaviors of the mentioned unary cost functions are illustrated in Figure 17. Note, however, that the figure only shows an illustration from a sample neighborhood window as the actual label space is 0 to 255. Also, we mainly use neighborhood window size of 5x5 in our experiment.

Like stereo, all terms in the energy function were computed prior to energy minimization and then used throughout the experiment. The parameter combination used in each setting is the one that has achieved highest PSNR[14] computed from MQPBO results with respect to the ground truth images. The initial labeling for each pixel of α-expansion algorithm is taken directly from the intensity value of the input noisy image.

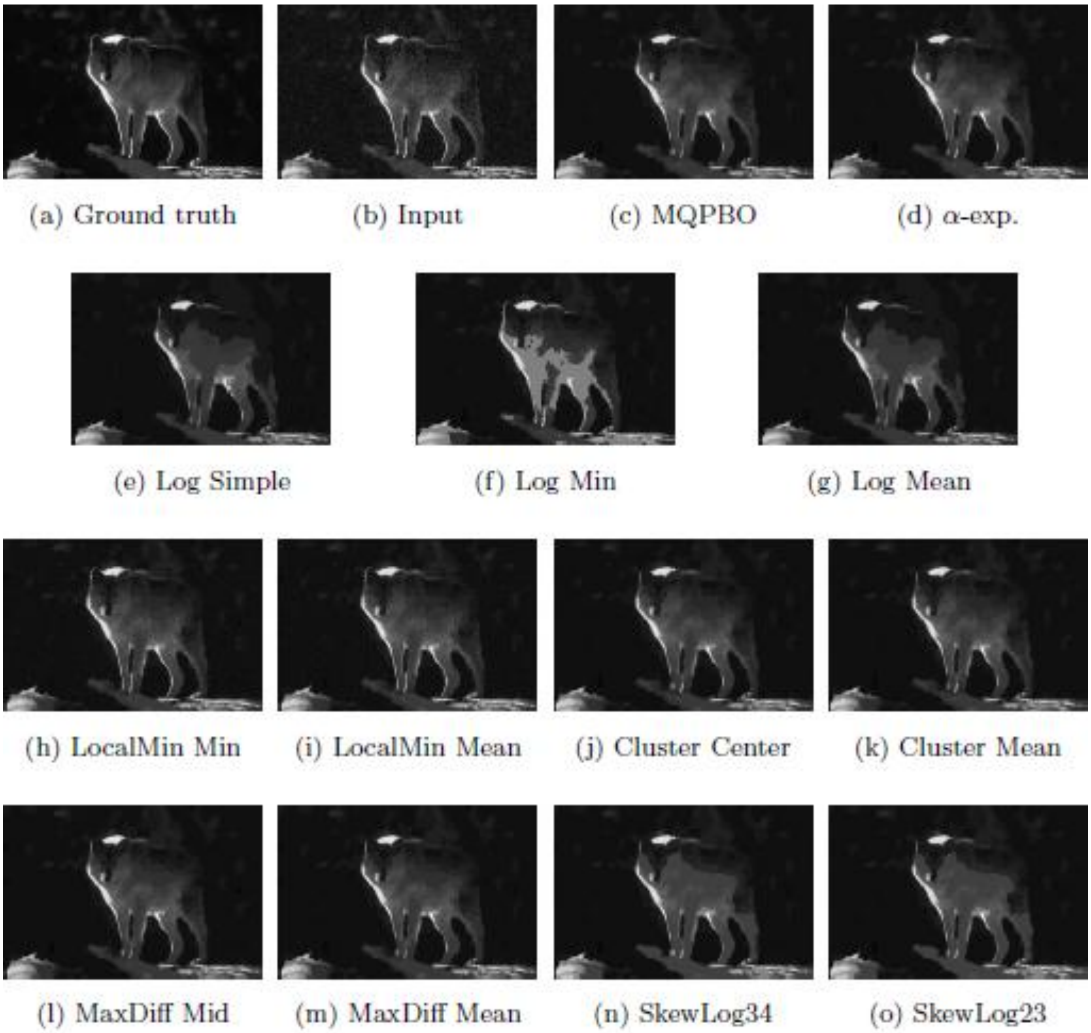

(a) 0072_35 orig (b) capricorn orig (c) mushroom orig

(d) 0072_35 input (e) capricorn input (f) mushroom input

Figure 18: Examples of original and input images from the datasets (From left to right: taken from [Estrada et al. 2009], [Pletscher et al. 2011], and [Mairal et al. 2008].)

[14] Peak Signal-to-noise Ratio is calculated from $\mathrm{PSNR} = 10\log\left(\frac{max\_intensity^2}{\mathrm{MSE}}\right)$ where $max\_intensity$ is the maximum possible intensity value of the input and MSE is the Mean Squared Error of the denoising result with respect to the original image, which is calculated from $\mathrm{MSE} = \frac{1}{width \times height}\sum_{x=0}^{width-1}\sum_{y=0}^{height-1}\left(original(x,y) - result(x,y)\right)^2$.

The datasets we used are mostly from the Image Denoising Benchmark[15] [Estrada, *et al.* 2009]. Others are from [Pletscher, *et al.* 2011] (capricorn, arch, foxes, elephant, etc.) and [Mairal, *et al.* 2008] (castle, mushroom, etc.). The benchmark in [Estrada, *et al.* 2009] provides multiple noise levels of each image indicated by the standard deviation of Gaussian noise added. We have converted the images to grayscale to use in the experiment. Figure 18 shows some examples of the original and input noisy images from the datasets.



(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

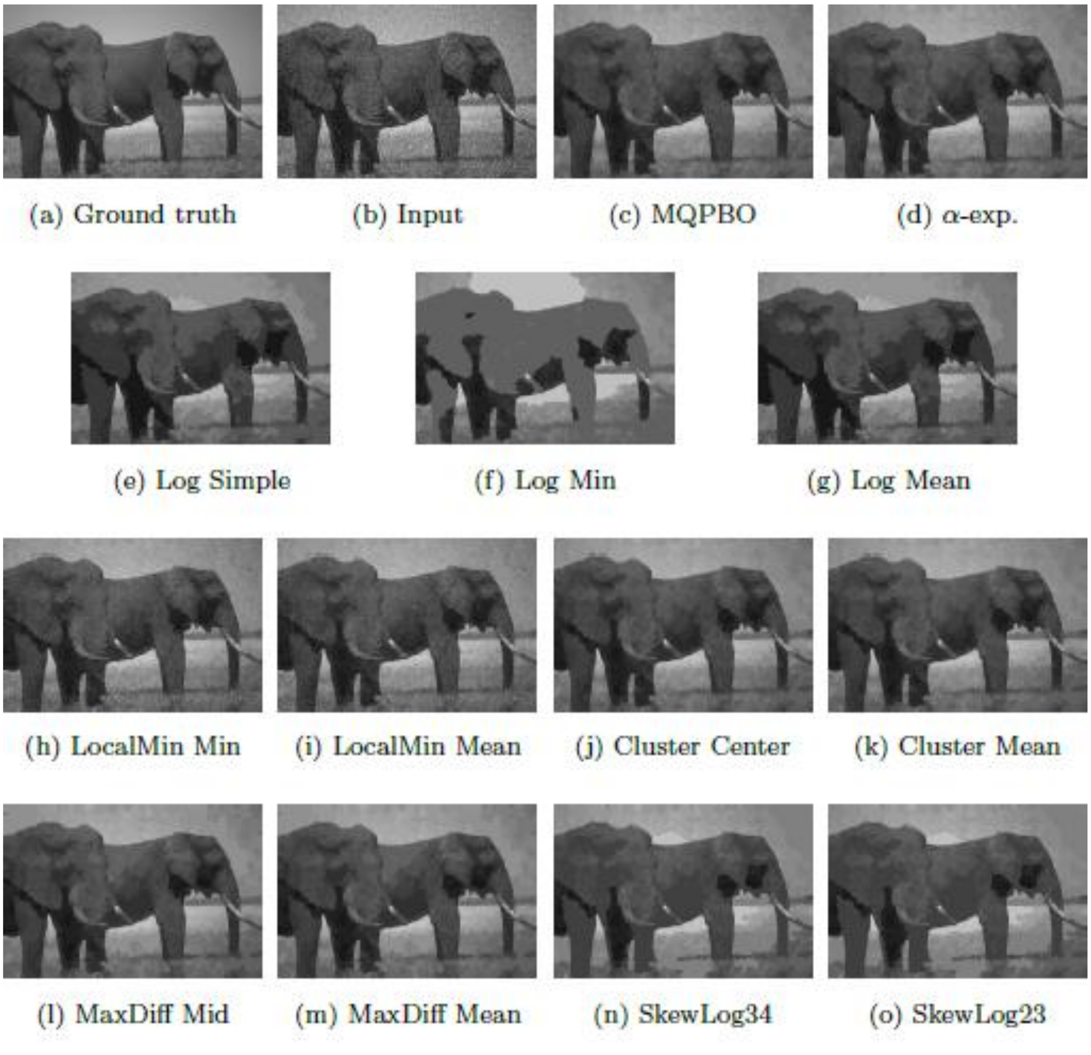
(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 19: Denoising results of "Wolf" image

[15] URL: www.cs.utoronto.ca/~strider/Denoise/Benchmark/

The denoising results of "Wolf" at SD=35 (0246_35) (dataset from [Estrada, *et al.* 2009]) are shown in Figure 19. The results are from using sum of square-root of absolute differences cost and inverse smoothness prior. The artifacts in **Log** techniques are the most noticeable, followed by **SkewLog** techniques. The rest are more subtle and have to be inspected by PSNR results.



(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 20: Denoising results of "Elephant" image

Figure 20 shows the denoising results of "Elephant" (dataset from [Pletscher, et al. 2011]). Again, the behaviors of the techniques are the same as the previous case with **Log** techniques having the most noticeable artifacts followed by **SkewLog**. This input image, however, exposes more artifacts in the results since it has lower PSNR than that of the previous one.

Table 4: PSNR results of "Wolf" and "Elephant" datasets

Wolf **Techniques**	**PSNR**		**Elephant** **Techniques**	**PSNR**
Input	26.6269		Input	20.2098
MQPBO	30.6166		MQPBO	28.6284
AlphaExp	30.6165		AlphaExp	28.6275
Log Simple	29.8146		Log Simple	26.9934
Log Min	29.5890		Log Min	26.2698
Log Mean	29.8158		Log Mean	27.0137
LocalMin Min	30.2596		LocalMin Min	28.1523
LocalMin Mean	30.3612		LocalMin Mean	28.1599
Cluster Center	30.6130		Cluster Center	28.6236
Cluster Mean	30.6080		Cluster Mean	28.6117
MaxDiff16 Mid	30.5866		MaxDiff16 Mid	28.6096
MaxDiff16 Mean	30.6138		MaxDiff16 Mean	28.6246
SkewLog 3/4	30.1539		SkewLog 3/4	27.9068
SkewLog 2/3	30.1394		SkewLog 2/3	27.9065

The PSNR results of "Wolf" and "Elephant" in the previous figures are shown in Table 4. In both cases, the PSNR results reveal that the relative qualities of the results from techniques being compared agree with that of the stereo benchmark from section 4.2. **Log** techniques give the worst PSNR with **Log Min** having the lowest values. **SkewLog** techniques are next with **SkewLog34** having higher PSNR than **SkewLog23**. Next are the **LocalMin** techniques with **LocalMin Min** giving worse PSNR than **Mean**. Results from **Cluster Center** and **MaxDiff Mean** are nigh on par with the baseline **MQPBO** and **AlphaExp** techniques with **MaxDiff Mean** having slightly better PSNR than **Cluster Center**.

The speedup, energy ratio, and error results from all settings of image denoising experiment are shown in Table 5. Overall, the relative performances of the techniques do agree with those from the stereo experiment. The trade-off between speed and quality can be seen by comparing **Log** and **SkewLog** techniques. Among our proposed label-wise grouping techniques, **LocalMin** techniques still give poorest quality results but faring well in terms of speed. **Cluster Center**, in contrast, does well in terms of energy and PSNR but provides lower speedup. **MaxDiff** techniques still perform well on all metrics being compared with **MaxDiff16** being the best among all preset number of subgroups and **MaxDiff16 Mean** being better than both **LocalMin Mean** and **Cluster Center**.

Table 5: Speedup, energy ratio and error results from image denoising experiment

Technique	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**	**Technique**	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**
MQPBO	1	---	26.656	MaxDiff8 Mid	9.59	+11.42%	26.266
AlphaExp	0.979	+0.114%	26.654	MaxDiff8 Mean	10.1	+7.88%	26.343
Log Simple	18.9	+81.57%	24.595	MaxDiff16 Mid	10.9	+6.11%	26.499
Log Min	16.1	+113.95%	24.107	MaxDiff16 Mean	11.1	+3.95%	26.650
Log Mean	18.1	+77.74%	24.649	MaxDiff32 Mid	9.68	+9.20%	26.432
LocalMin Min	11	+28.29%	26.018	MaxDiff32 Mean	10.1	+7.66%	26.519
LocalMin Mean	9.97	+17.73%	26.136	SkewLog 3/4	14.1	+61.51%	25.652

Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	PSNR	Technique	Speedup wrt. MQPBO	Energy ratio wrt. MQPBO	PSNR
Cluster Center	9.38	+4.52%	26.645	SkewLog 2/3	15.6	+63.93%	25.093
Cluster Mean	8.31	+8.65%	26.471				

4.4 Inpainting Benchmarks

There are many circumstances in which parts of an image may be lost or corrupted. Films may deteriorate, photographs may crack, or blocks may be lost in the coding and transmission of images. Some artifacts may even be intentionally added, such as timestamps and watermarks. The problem of image inpainting accepts as input an image to be inpainted and a mask of the same size specifying the state of each pixel as "known" or "unknown".

Unlike denoising in which the intensity values of the pixels are observed with possible errors, the case where parts of the image are missing poses a different challenge to the problem of estimating the true observations. Given a pixel that is obscured or damaged, its intensity is estimated by the intensity of its neighbors. This common practice of using a kernel of fixed size as neighborhood window works adequately when the pixel has some neighbors with correct intensity values but is not applicable when the pixel to be estimated is too far for the kernel to reach. In terms of energy function, this means that its unary potential would be equal for all possible intensity labels to reflect the fact that there is no information in the pixel itself and the estimation must rely on the pairwise potentials for the information to propagate through.

We have also experimented with using distance transform to calculate pairwise potential. This is to reflect the fact that unknown pixel situated nearer to known pixels should be penalized more for having different intensity value from its neighbors than the further situated counterparts. Given the input mask, the distance from each unknown pixel to the nearest known pixel is calculated. This distance is then used as input to calculate smoothness priors for pairwise potentials. In addition, we also use the direction to the nearest known pixel to specify the strength of the pairwise potential in each direction.

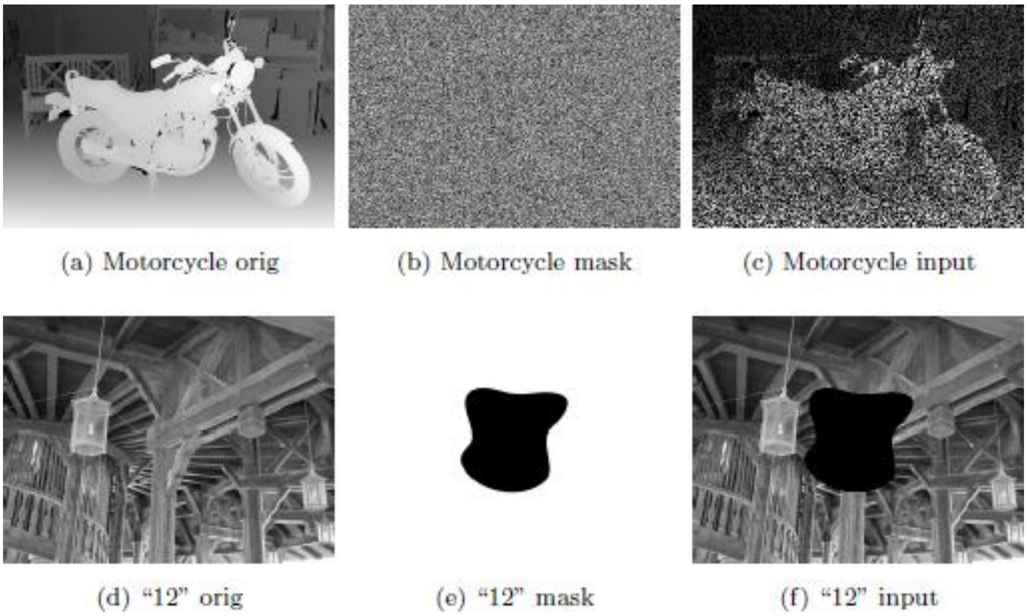


Figure 21: Examples of original, mask, and input images from the datasets (Top row: from TUM-IID; Bottom row: from Depth Inpainting database)

The datasets we used in our experiment are from the TUM-Image Inpainting Database[16] [Tiefenbacher *et al.* 2015] and the Depth Inpainting database[17] [Xue *et al.* 2017]. The TUM-IID contains natural scene images that are diverse in terms of texture and structure as well as different masks to specify target regions to be inpainted. The Depth Inpainting database consists of depth images converted from ground truth disparity maps, the masks specifying missing depth values and the damaged images. Figure 21 shows examples of the original, mask, and input images from the datasets. Note that the PSNR values reported are computed over only the unknown regions, which means that the inputs would always have PSNR of 0 dB.

[16] URL: www.mmk.ei.tum.de/tumiid/

[17] URL: www.cad.zju.edu.cn/home/dengcai/Data/depthinpaint/DepthInpaintData.html

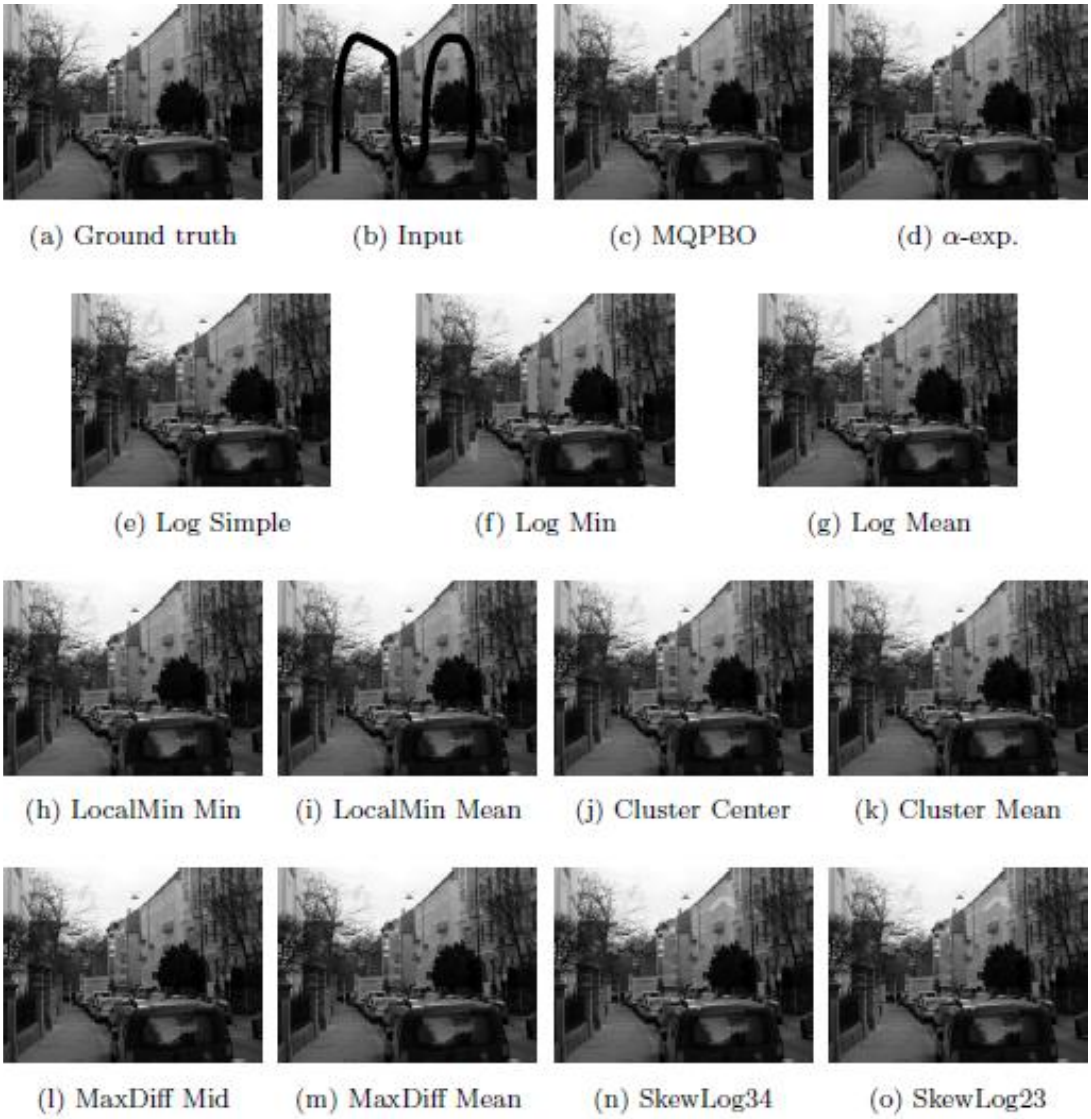
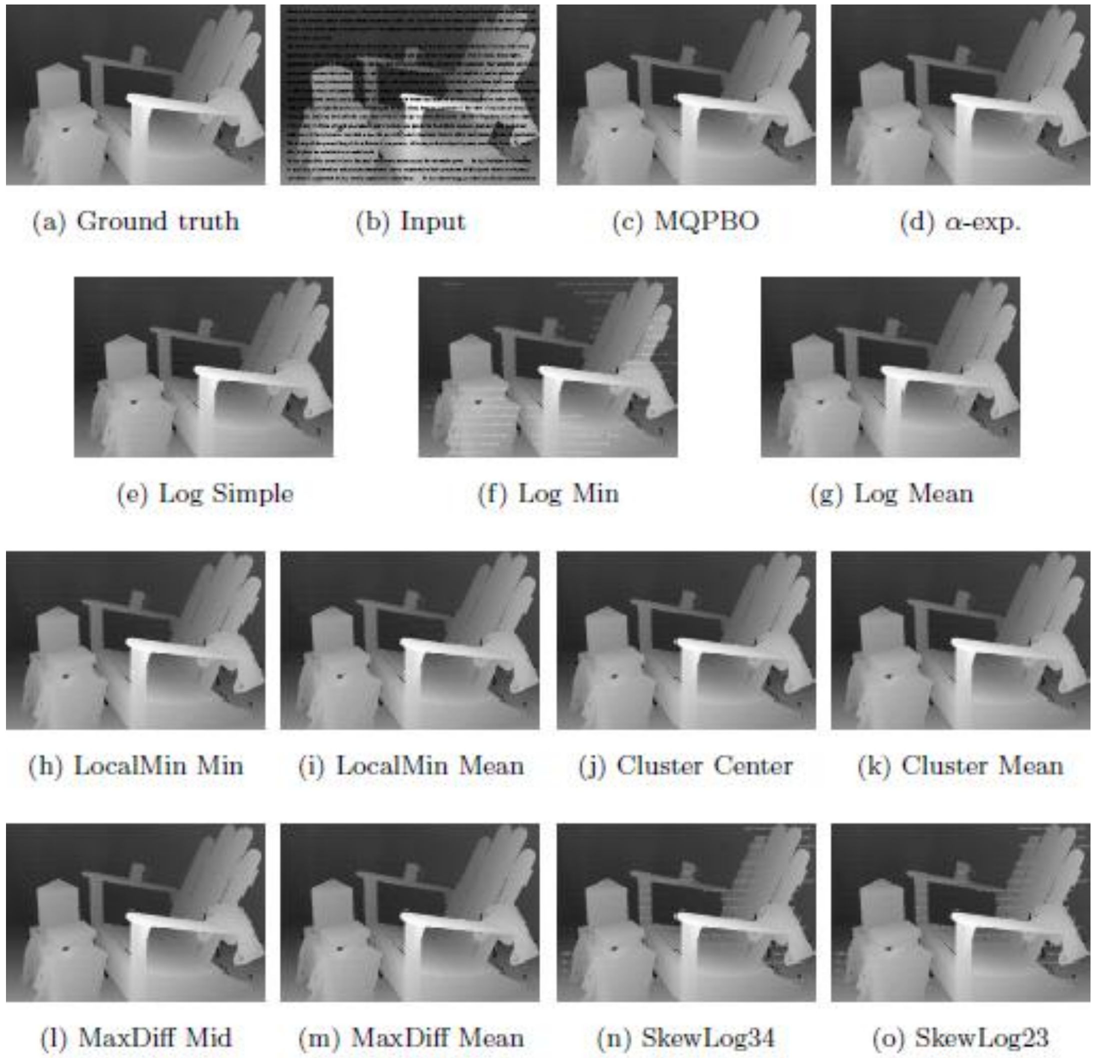


Figure 22: Inpainting results of "16" from the TUM-IID dataset

Figure 22 shows the inpainting results of "16" from the TUM-IID dataset and Figure 23 shows the results of "adi" from the Depth Inpainting database. All of the results are from using sum of square-root of absolute differences cost and inverse smoothness prior. As with stereo and image denoising, artifacts in **Log** and **SkewLog** techniques are more easily to be seen than those in the other techniques.


(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 23: Inpainting results of "adi" from the Depth Inpainting database

Their PSNR results are shown in Table 6. The relative quality of the inpainting results from most of the techniques does agree with stereo and image denoising except for **Cluster** techniques, which perform worse than **LocalMin** (and, also, worse than **MaxDiff**). This may seem counterintuitive at first but, as mentioned before, **Cluster** does not group labels with equal unary potentials which should have made those labels behave like they were being solved with **MQPBO**. Instead, because the unknown pixels close to the border of the mask were still able to calculate potential based on some of their neighbors, forcing the pixels that are farther to make decision at the root worsens the quality of the results rather obtrusively. The results from delaying the ungrouped labels to make decision at the 0 th level are shown in the table as **Cluster(delay) Center** and **Mean**, which now give better PSNR than **MaxDiff Mean**. Note that the image results shown in the figures are from the delayed versions of **Cluster**.

Table 6: PSNR results of "16" and "adi" datasets

16 **Techniques**	**PSNR**		**adi** **Techniques**	**PSNR**
MQPBO	34.604		MQPBO	27.104
AlphaExp	34.607		AlphaExp	27.088
Log Simple	30.687		Log Simple	26.645
Log Min	28.78		Log Min	26.567
Log Mean	30.702		Log Mean	26.655
LocalMin Min	33.736		LocalMin Min	27.033
LocalMin Mean	33.737		LocalMin Mean	27.047
Cluster Center	33.4807		Cluster Center	26.986
Cluster Mean	33.3923		Cluster Mean	26.876
MaxDiff16 Mid	34.597		MaxDiff16 Mid	27.07
MaxDiff16 Mean	34.601		MaxDiff16 Mean	27.076
SkewLog 3/4	32.712		SkewLog 3/4	26.884
SkewLog 2/3	31.844		SkewLog 2/3	26.837
Cluster(delay) Center	34.604		Cluster(delay) Center	27.089
Cluster(delay) Mean	34.579		Cluster(delay) Mean	27.088

In regions where the unary potentials of all labels are equal, **MaxDiff** techniques create groups of equal size since all the potential differences are zero. On the other hand, **LocalMin** uses its group size constraint to build the hierarchy. Nevertheless, the groups created by **MaxDiff16** and **LocalMin** are not the same. **MaxDiff**'s strategy is to split the largest group nearest to its middle so **MaxDiff16** would create 16 groups of size 16. **LocalMin**, if used group size constraint of 16 as it gives the best performance in **MaxDiff**, would first choose the middle label as the local minimum and then expand on both sides to reach the size constraint so, for label space [0,255], the first group created would be [120,135]. Next, **LocalMin** would repeat the process for [0,119] and [136,255] and create [52,67] and [188,203]. The process would go on until all groups are of size not larger than 16 and would result in more than 16 groups with different sizes[18]. We use group size constraint of 18 in our experiment as its 15 resulting groups most resemble those built by **MaxDiff16**.

Table 7: Speedup, energy ratio and error results from inpainting experiment

Technique	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**	**Technique**	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**
MQPBO	1	---	24.825	MaxDiff8 Mid	10.3	+9.62%	24.466
AlphaExp	0.978	+0.248%	24.823	MaxDiff8 Mean	10.8	+6.48%	24.540
Log Simple	18.8	+58.09%	23.827	MaxDiff16 Mid	10.9	+4.77%	24.681

[18] The groups for **MaxDiff16** are {[0,15], [16,31], [32,47], [48,63], [64,79], [80,95], [96,111], [112,127], [128,143], [144,159], [160,175], [176,191], [192,207], [208,223], [224,239], [240,255]} while **LocalMin** would yield {{0}, [1,16], {17}, [18,33], {34}, [35,50], {51}, [52,67], {68}, [69,84], {85}, [86,101], {102}, [103,118], {119}, [120,135], {136}, [137,152], {153}, [154,169], {170}, [171,186], {187}, [188,203], {204}, [205,220], {221}, [222,237], {238}, [239,254], {255}} with maximum group size 16 and {[0,15], [16,33], [34,49], [50,67], [68,83], [84,101], [102,118], [119,136], [137,152], [153,170], [171,186], [187,204], [205,220], [221,238], [239,255]} with maximum group size 18.

Technique	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**	**Technique**	**Speedup wrt. MQPBO**	**Energy ratio wrt. MQPBO**	**PSNR**
Log Min	18.5	+58.77%	23.352	MaxDiff16 Mean	11.3	+3.48%	24.821
Log Mean	19	+57.43%	23.687	MaxDiff32 Mid	10.3	+8.02%	24.605
LocalMin Min	11.1	+6.46%	24.423	MaxDiff32 Mean	10.6	+6.81%	24.701
LocalMin Mean	11	+3.71%	24.434	SkewLog 3/4	18.2	+47.76%	24.292
Cluster Center	2.04	+1.08%	24.821	SkewLog 2/3	18.3	+50.72%	24.242
Cluster Mean	2.03	+2.02%	24.800				

Table 7 summarizes the speedup, energy ratio, and error results from all settings of inpainting experiment. The key difference between the results here and those in the previous benchmark applications is the existence of regions where the random variables have equal unary potentials for all labels. The trade-off between **Log** and **SkewLog** persists but with smaller gaps since both **SkewLog** techniques split groups at the middle the same way as **Log** in equi-potential regions. Also in these regions, **LocalMin** and **MaxDiff** techniques behave similarly with slightly different resulting groups as previously discussed. It can be seen that **LocalMin**'s 15 subgroups performs marginally worse than **MaxDiff16**. Using **Min** and **Mid** still give poorer quality results than **Mean**, with **Mid** being a better group representative potential than **Min**. The results shown for both **Cluster** techniques are the "delay" versions. **Cluster Center** now has only 1% increase in average energy and virtually the same PSNR as the baseline techniques since it leaves the labels as they were without being grouped. This, of course, means that both **Cluster** techniques now give considerably less gain in speedup.

4.5 Discrete Energy Minimization Benchmarks

We have also experimented with a database of discrete energy minimization problems from the OpenGM Benchmark[19] [Kappes, *et al.* 2015] which contains datasets from various applications from both within and outside the field of computer vision. We selected mrf-inpainting and mrf-stereo benchmarks for problems with linearly ordered label space. Unlike the stereo benchmarks in section 4.2 which has label space size of 256, mrf-stereo benchmark has label space sizes between 16 and 60. We have also compared the results of applications where the labels do not have natural ordering structure that represents physical quantity, namely, image-seg and protein-folding-pdb benchmarks. In image-seg benchmark, the random variables are segmented superpixels in the input image and the label space contains as many labels as there are superpixels, making a rather large label space size for some instances. The protein-folding-pdb benchmark refers to protein folding side-chain prediction [Yanover *et al.* 2008] and the constructed models are fully connected and have quite large label spaces. We have chosen these applications because of their large sizes of label spaces.

Here we have experimented with using LBP and TRWS as the underlying optimization algorithms and, as such, they are the baseline used to calculate the speedup and energy ratio shown in the tables. Because of the irregular label space, we have used the square root of label space size as the number of subgroups for **MaxDiff**. It can be seen that changing the optimizers does have effect on speedup and energy results.

Table 8 and Table 9 summarize the results. For mrf-inpainting and mrf-stereo, the results are consistent with those in the previous sections. The speedup gains for mrf-stereo are less manifest here than in section 4.2 since the label spaces in this case are not full-sized.

[19] URL: hciweb2.iwr.uni-heidelberg.de/opengm/

Table 8: Speedup and energy ratio results of mrf-inpainting and mrf-stereo

	mrf-inpainting				**mrf-stereo**			
	LBP		**TRWS**		**LBP**		**TRWS**	
Technique	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**
Log Simple	12.2	+47.43%	15.9	+46.75%	3.01	+86.56%	4.51	+108.27%
Log Min	12.4	+52.36%	16.1	+49.80%	3.25	+90.98%	4.72	+123.01%
Log Mean	12.3	+49.67%	16	+45.93%	3.18	+79.71%	4.7	+107.81%
LocalMin Min	8.12	+22.37%	9.91	+31.87%	2.84	+39.84%	3.71	+20.85%
LocalMin Mean	7.99	+17.09%	9.89	+30.34%	2.74	+20.90%	3.44	+15.66%
Cluster Center	6.3	+4.09%	7.89	+4.56%	2.01	+6.11%	2.61	+4.54%
Cluster Mean	6.18	+4.23%	7.67	+7.98%	2	+7.83%	2.27	+7.61%
MaxDiff Mid	7.89	+13.95%	9.87	+13.91%	2.82	+5.25%	3.52	+5.28%
MaxDiff Mean	8.13	+4.05%	9.97	+4.24%	2.85	+4.36%	3.69	+4.59%
SkewLog 3/4	10.5	+33.78%	13.5	+34.90%	2.93	+59.55%	4.33	+70.42%
SkewLog 2/3	10.9	+37.31%	14	+37.21%	2.97	+66.85%	4.37	+74.17%

For image-seg and protein-folding-pdb in which the label spaces are unstructured, both speedup and energy ratio results of the techniques still exhibit the same relative quality as in the linearly ordered label space but more erratically. The increases in energy for image-seg, though higher than linearly ordered label space cases, are still not above 10% for **MaxDiff**. **Cluster**, in this case, does not take effect since all variables have no unary term. For protein-folding-pdb, however, the lowest increase in energy is still almost 30%. Comparing the two, image-seg, while having as many labels as variables, can be segmented adequately when grouped since hierarchical label grouping still allows adjacent superpixels to congregate and differentiate. The same cannot be accomplished, at least not satisfactorily, for protein-folding-pdb.

Table 9: Speedup and energy ratio results of image-seg and protein-folding-pdb

	image-seg				**protein-folding-pdb**			
	LBP		**TRWS**		**LBP**		**TRWS**	
Technique	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**	**Spd up**	**Energy ratio**
Log Simple	3.94	+69.35%	6.4	+53.71%	10.3	+114.94%	12.2	+140.45%
Log Min	3.95	+72.04%	6.28	+54.99%	10.4	+118.49%	12.4	+147.13%
Log Mean	3.94	+63.90%	6.31	+53.77%	10.3	+114.17%	12.4	+140.39%
LocalMin Min	3.83	+16.36%	5.32	+19.24%	7.06	+59.04%	8.07	+62.16%
LocalMin Mean	3.8	+8.37%	5.25	+8.93%	7.02	+46.85%	8.07	+44.23%
Cluster Center	1	0.00%	1	0.00%	5.11	+35.36%	5.73	+29.14%
Cluster Mean	1	0.00%	1	0.00%	5.05	+40.60%	5.6	+37.32%

Technique	image-seg				protein-folding-pdb			
	LBP		TRWS		LBP		TRWS	
	Spd up	Energy ratio	Spd up	Energy ratio	Spd up	Energy ratio	Spd up	Energy ratio
MaxDiff Mid	3.73	+11.67%	5.23	+15.70%	6.84	+44.97%	7.63	+49.88%
MaxDiff Mean	3.83	+8.26%	5.26	+8.86%	7.04	+35.27%	8.06	+29.08%
SkewLog 3/4	3.87	+57.16%	6.22	+41.11%	8.88	+91.42%	10.5	+122.85%
SkewLog 2/3	3.88	+59.36%	6.25	+42.82%	9.13	+99.00%	10.6	+129.34%

4.6 Result Discussions

From the experimental results, it can be observed that the behavior and performance of the techniques being compared agree across the applications with linearly ordered label space used as benchmarks. **Log** techniques give the best speedups but poor energy ratios. In contrast, **SkewLog**, which also does binary subdivision but allows unequal group sizes to favor subdividing between labels with maximum difference in potentials, gives higher-quality results, which supports our thesis that incorporating information from the energy function into the subdivision process can lead to better quality of the results in terms of energy and error rate. **SkewLog**, however, oftentimes requires more levels in the hierarchy to be optimized which leads to smaller gains in speedup.

In general, using **Min** for group potential assignment provides poorer quality results than using **Mean**, both in **Log** and **LocalMin** as well as in our preliminary **SkewLog** results. This behavior is also consistent with the findings in [Lempitsky, *et* *al.* 2007], strongly indicating that lower bound approximation is not an effective choice for group potential assignment.

Among our proposed techniques, **LocalMin** favors speed over quality whereas **Cluster** does the opposite and provides quality while having less speedup gain. **MaxDiff**, however, performs respectably on both counts. **LocalMin** and **MaxDiff** give their respective best results when using **Mean** as group potential assignment. **Cluster**, on the other hand, does its best when using **Center**. The behavior is consistent across the benchmark applications, evidently suggesting that mean-shift centers are reliable as representative labels.

Between **LocalMin** and **MaxDiff**, their speedups are comparable on average but **LocalMin** gives inferior results in terms of energy and error despite both using **Mean** which gives their respective best. This is because the groups resulted from **LocalMin** can be more diverse since there is no guarantee that labels that fall into the same basin would be close in terms of potentials. Having labels with diverse potentials in the same group means that **Mean** is still not a good choice for group potential assignment even though it provides better results than **Min**.

MaxDiff can be regarded as an approximation of **Cluster** since both result in separating the pairs of adjacent labels with large unary potential differences. The main difference between the two strategies is that **Cluster** handles consecutive labels with close potentials at the top of the hierarchy while **MaxDiff** does so by delegating the overall workload between the levels. Dealing with variables having these equi-potential labels at the top, however, is not a good practice as shown empirically in the results. This is because doing so would result in more imbalanced workloads between the hierarchical levels, giving more workload to the root and therefore gaining less speedup. Also, forcing these variables to make irreversible decisions at the top while the rest still have not decided leaves no room for them to refine their decisions once their neighbors have finished making their decisions. This effect is most obvious in inpainting where delaying the processing of pixels situated farther into the unknown regions to the 0th level gives noticeably better PSNR and energy results. **Cluster(delay)**, nonetheless, still provides subpar speedup since the workloads between the levels are still imbalanced.

Among the different numbers of subgroups in **MaxDiff** techniques, **MaxDiff16** provides the best results in all measures. This is by no means a coincidence since all of the benchmark applications with linearly ordered label space have a label space size of 256, which makes 16 the choice of number of subgroups that would result in a most balanced two-level hierarchy. Using more subgroups can result in groups in which the labels have close unary potentials being subdivided which leads to less speedup and upper-level solution getting stuck in a local minimum while using fewer subgroups can group labels with more diverse potentials together which can mislead the group potential estimation in the upper-level optimization. Our preliminary experiments using a coarser label space size of 32 also had 6 being the best choice for number of subgroups followed closely by 5.

Another advantage of our methodology is the lowered memory requirement for calculation. While α-expansion does not face physical memory limit as it explores only a relatively much smaller fraction of the solution space, the memory occupancy of MQPBO method is generally much higher by several orders of magnitude and virtual memory swapping can have drastic effect on its running time. Our methodology's theoretical memory consumption, while still being significantly higher than that of α-expansion, is several orders of magnitude smaller than that of MQPBO method. For one random variable with 256 possible labels, the number of vertices constructed in max-flow-based energy minimization is 2 for QPBO-based α-expansion, 30 for MQPBO-based **MaxDiff16**, and 310 for MQPBO method. For a pairwise interaction between two variables each having 256 labels, the number of directed edges constructed is 4 for QPBO-based α-expansion, 900 for MQPBO-based **MaxDiff16**, and 260100 for MQPBO method.

In recent years, the combination of advances in learning methods and parallel computing devices together with the availability of easy-to-access large-scale datasets has made possible the rapid development in deep learning and convolutional neural networks (CNNs) for the use in many research areas including computer vision. For image denoising, the work of [Zhang *et al.* 2017] uses CNN for residual learning which outputs an estimate residual mapping to be subtracted from the noisy input to obtain the latent clean result. The receptive field size has to be quite large (35x35) and the network has to be rather deep (17 and 20) but the overall computation can be sped up by GPU, compared with nonlocal self-similarity (NSS) models which are popular in state-of-the-art approaches and have presented high denoising quality but require complex optimization steps that can be more time-consuming. Generative adversarial networks (GANs) are used in [Yeh *et al.* 2017] for image inpainting. Given a fixed-size corrupted image, their work tries to recover the encoding closest to the image and then uses the generator trained from GAN to generate the unknown region. For certain problems such as stereo correspondence, the use of deep learning is still mainly for calculating matching cost to be used in random field formulation [Chen *et al.* 2015, Luo *et al.* 2016], which stresses the importance of random field for handling problematic regions such as those with occlusions, repetitive patterns or lack of distinctive features. Overall, the use of image-wide global information in deep learning and CNNs is still somewhat limited. In order to capture enough spatial information to perform computer vision tasks, either random field formulation is still needed or the networks have to be very deep so that the corresponding effective receptive field sizes are adequately large. We expect to see more of the interplay between deep learning architecture and random field formulation in the near future.

Chapter 5: Conclusions

5.1 Dissertation Summary

Random field formulation has been prominent in computer vision research and related area due to the ability to intuitively incorporate spatial relationships and visual contexts with local information. Its phenomenal success since its introduction in low-level vision tasks has drawn in researchers and, through the test of time, it has been recognized to be a powerful framework for solving a wide and diverse range of computer vision applications from image denoising, multi-view reconstruction, and motion estimation to object recognition and scene understanding. As a result, random field optimization or finding a MAP solution for a random field and its equivalent form, discrete energy minimization, have been of central interest in computer vision research community for over a decade.

As the technology progresses, the demand for computer vision has grown to be greater and more diverse than ever. Larger problem instances mean larger sets of random variables and larger label spaces are involved. This, unfortunately, means that solving for solutions, even in an approximated sense, can be impractical since the computational complexity grows fast with the size of the problem. This plus the abundance of labeling problems in computer vision make it very important to develop tractable algorithms that can handle diverse classes of problems as creating more models for problems will not lead to anything worthwhile if reasonable labelings cannot be found in practice.

Our main contribution to random field optimization is the introduction of local label hierarchy for hierarchical energy minimization. We focus on problems in which the label space has a natural linear ordering structure that represents physical quantity and utilize this characteristic of the underlying labeling problems, which is the key that has enabled us to circumvent exhaustive search of the solution space and obtain a more computationally efficient scheme for energy minimization. We give notations and definitions for local label hierarchy as well as generalize the definition of discrete energy function to include sets of labels in the domain. Three techniques for label-wise grouping are proposed: local minimum search, cluster analysis, and maximum-difference subdivision, as well as one generalization of algorithms of logarithmic complexity. Also, heuristics for assigning group potentials are discussed.

Our methodology was tested with a number of computer vision applications with linearly structured label spaces as well as problems with unstructured label spaces. Comparing **Log** with **SkewLog** corroborates our thesis that including energy information into the binary subdivision process leads to better energy and error results. The three contenders among our proposed label-wise grouping techniques are **LocalMin Mean**, **Cluster Center**, and **MaxDiff Mean** with **MaxDiff Mean** being the most competitive, providing approximately an order of magnitude speedup with less than 5% increase in energy in all benchmark applications. Overall, grouping labels with close potentials together gives better results due to better estimates of group potentials and the key to performing well in terms of both speed and quality is to balance the workload across the hierarchical levels.

5.2 Open Questions

While we have described our efforts in speeding up the energy minimization process in this dissertation, there are still exciting challenges that lay ahead. Potential improvement and extension for this work are listed in this section as well as some open questions.

- Unlike having the same hierarchy for every variable, our local label hierarchy presents quite a challenge for learning-based group potential assignment as the parameters generally depend on both group size and level. Cluster analysis has the advantage of having a natural representative label for each group which gives decent results when used. Finding a better way to assign group potential for maximum-difference subdivision could potentially give even higher-quality results.

- While using the square root of the label space size as the predetermined number of subgroups for maximum-difference subdivision empirically provides respectable speedup and high-quality results in terms of energy and error than using mean-shift cluster analysis to automatically find the number of groups from the energy function, there is still a question of how to automatically determine the number of subgroups that gives the best trade-off between speedup and result quality.

- It has been shown in the literature that random field optimization for problems with structured label spaces can be made more efficiently by hierarchically solving for solutions. The question remains, however, whether such structure can be detected or inferred automatically from the energy functions. In the most extreme sense, can any label space be automatically arranged and grouped into hierarchy in such way that efficient hierarchical energy minimization can be utilized?

5.3 Final Remarks

As pointed out before by many, having found a configuration with the lowest energy does not automatically means that it is the best in terms of quality. In fact, comparing the energy from the minimum energy configuration with that from the ground truth often reveals that the ground truth considerably has higher energy. The obvious solution to this is, of course, to create a more accurate formulation that better models the labeling problem. Researchers, of course, need to provide results. This drive to have tangible outcomes, perhaps, implies that we as a whole may have been biased by algorithms that are available and perform well, which makes problem formulation incline toward more obtainable goals rather than more intuitively appealing models. As random field formulation was once thought to be too intractable for practical use, we hope that more efficient algorithms would encourage more complex and elegant problem formulations which, in turn, would loop back and drive the research community toward the development of new and even more efficient algorithms.

REFERENCES

AHUJA, R., MAGNANTI, T. and ORLIN, J. *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, (1993).

AHUJA, R. K. and ORLIN, J. B. *A Fast and Simple Algorithm for the Maximum Flow Problem*, (1989).

AHUJA, R. K., ORLIN, J. B. and TARJAN, R. E. Improved Time Bounds for the Maximum Flow Problem. *SIAM Journal on Computing* 18 (1989), 939-954.

ALON, N. Generating pseudo-random permutations and maximum flow algorithms. *Information Processing Letters* 35 (1990), 201-204.

ANDRES, B., KAPPES, J. H., BEIER, T., KÖTHE, U. and HAMPRECHT, F. A. *The Lazy Flipper: Efficient Depth-Limited Exhaustive Search in Discrete Graphical Models*, Springer Berlin Heidelberg, (2012).

BERGTHOLDT, M., KAPPES, J., SCHMIDT, S. and SCHNÖRR, C. A Study of Parts-Based Object Class Detection Using Complete Graphs. *International Journal of Computer Vision* 87 (2010), 93.

BESAG On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B Methodological* 48 (1986), 259-302.

BIRCHFIELD, S. and TOMASI, C. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), 401-406.

BISHOP, C. M. *Pattern Recognition and Machine Learning*, Springer, (2006).

BOROS, E. and HAMMER, P. L. Pseudo-Boolean optimization. *Discrete Applied Mathematics* 123 (2002), 155-225.

BOYKOV, Y. and FUNKA-LEA, G. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision* 70 (2006), 109-131.

BOYKOV, Y. and KOLMOGOROV, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26 (2004), 1124 -1137.

BOYKOV, Y. and VEKSLER, O. *Graph Cuts in Vision and Graphics: Theories and Applications*, Springer US, (2006).

BOYKOV, Y., VEKSLER, O. and ZABIH, R. *Markov random fields with efficient approximations*. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings (1998), 648-655.

BOYKOV, Y., VEKSLER, O. and ZABIH, R. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23 (2001), 1222 -1239.

CHEN, Z., SUN, X., WANG, L., YU, Y. and HUANG, C. *A Deep Visual Correspondence Embedding Model for Stereo Matching Costs*. 2015 IEEE International Conference on Computer Vision (ICCV) (2015), 972-980.

CHENG, Y. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (1995), 790-799.

CHERIYAN, J. and HAGERUP, T. *A randomized maximum-flow algorithm*. Foundations of Computer Science, 1989., 30th Annual Symposium on (1989), 118-123.

CHERIYAN, J. and HAGERUP, T. A Randomized Maximum-Flow Algorithm. *SIAM Journal on Computing* 24 (1995), 203-226.

CHERIYAN, J., HAGERUP, T. and MEHLHORN, K. An $o(n^3)$-Time Maximum-Flow Algorithm. *SIAM Journal on Computing* 25 (1996), 1144-1170.

CHERKASSKY, B. V. An algorithm for constructing a maximal flow through a network requiring O (n2√ p) operations. *Mathematical Methods for Solving Economic Problems* 7 (1977), 117–126.

COMANICIU, D. and MEER, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), 603-619.

CONEJO, B., KOMODAKIS, N., LEPRINCE, S. and AVOUAC, J. P. *Inference by Learning: Speeding-up Graphical Model Optimization via a Coarse-to-Fine Cascade of Pruning Classifiers*, Curran Associates, Inc., (2014).

CREMERS, D. and GRADY, L. *Statistical Priors for Efficient Combinatorial Optimization Via Graph Cuts*, Springer Berlin Heidelberg, (2006).

DAHLHAUS, E., JOHNSON, D. S., PAPADIMITRIOU, C. H., SEYMOUR, P. D. and YANNAKAKIS, M. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing* 23 (1994), 864-894.

DANTZIG, G. B. Application of the simplex method to a transportation problem. *Activity analysis of production and allocation* 13 (1951), 359–373.

DELONG, A., GORELICK, L., VEKSLER, O. and BOYKOV, Y. Minimizing Energies with Hierarchical Costs. *International Journal of Computer Vision* 100 (2012), 38-58.

DINIC, E. A. *Algorithm for solution of a problem of maximum flow in networks with power estimation*. Soviet Math. Dokl (1970), 1277–1280.

DINIC, E. A. Metod porazryadnogo sokrashcheniya nevyazok i transportnye zadachi. *Issledovaniya po Diskretnoi Matematike* (1973), 46–57.

EDMONDS, J. and KARP, R. M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *J. ACM* 19 (1972), 248–264.

ESTRADA, F., FLEET, D. and JEPSON, A. *Stochastic Image Denoising*. Proceedings of the British Machine Vision Conference, BMVA Press (2009), 117.111-117.111.

FELZENSZWALB, P. F. and HUTTENLOCHER, D. P. Pictorial Structures for Object Recognition. *International Journal of Computer Vision* 61 (2005), 55-79.

FORD, L. R. and FULKERSON, D. R. Maximal flow through a network. *Canadian Journal of Mathematics* 8 (1956), 399–404.

GALIL, Z. and NAAMAD, A. An O(EVIog2V) algorithm for the maximal flow problem. *Journal of Computer and System Sciences* 21 (1980), 203-217.

GOLDBERG, A. V. *Recent developments in maximum flow algorithms*, Springer Berlin Heidelberg, (1998).

GOLDBERG, A. V. and RAO, S. *Beyond the flow decomposition barrier*. Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on (1997), 2-11.

GOLDBERG, A. V. and RAO, S. Beyond the Flow Decomposition Barrier. *J. ACM* 45 (1998), 783–797.

GOLDBERG, A. V. and TARJAN, R. E. *A new approach to the maximum flow problem*. Proceedings of the eighteenth annual ACM symposium on Theory of computing, ACM (1986), 136–146.

GOLDSCHLAGER, L. M., SHAW, R. A. and STAPLES, J. The maximum flow problem is log space complete for P. *Theoretical Computer Science* 21 (1982), 105-111.

GREENLAW, R., HOOVER, H. J. and RUZZO, W. L. *Limits to parallel computation: P-completeness theory*, Oxford university press Oxford, (1995).

HAMMER, P., HANSEN, P. and SIMEONE, B. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical Programming* 28 (1984), 121-155.

HIRSCHMULLER, H. *Accurate and efficient stereo processing by semi-global matching and mutual information*. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (2005), 807 - 814 vol. 802.

HIRSCHMULLER, H. and SCHARSTEIN, D. *Evaluation of Cost Functions for Stereo Matching*. IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07 (2007), 1-8.

ISHIKAWA, H. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), 1333 - 1336.

KAPPES, J. H., ANDRES, B., HAMPRECHT, F. A., SCHNÖRR, C., NOWOZIN, S., BATRA, D., KIM, S., KAUSLER, B. X., KRÖGER, T., LELLMANN, J., KOMODAKIS, N., SAVCHYNSKYY, B. and ROTHER, C. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *International Journal of Computer Vision* (2015), 1-30.

KAPPES, J. H., SAVCHYNSKYY, B. and SCHNORR, C. *A bundle approach to efficient MAP-inference by Lagrangian relaxation*. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012), 1688-1695.

KARGER, D. R., KLEIN, P., STEIN, C., THORUP, M. and YOUNG, N. E. *Rounding Algorithms for a Geometric Embedding of Minimum Multiway Cut*. Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, ACM (1999), 668–678.

KARP, R. M. *Reducibility among Combinatorial Problems*, Springer US, (1972).

KARZANOV, A. V. *Determining the maximal flow in a network by the method of preflows*. Soviet Mathematics Doklady (1974), 434–437.

KIM, J., KOLMOGOROV, V. and ZABIH, R. *Visual correspondence using energy minimization and mutual information*. Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on (2003), 1033 -1040 vol.1032.

KINDERMANN, R. and SNELL, J. L. *Markov Random Fields and Their Applications*, American Mathematical Society, (1980).

KING, V., RAO, S. and TARJAN, R. *A Faster Deterministic Maximum Flow Algorithm*. Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics (1992), 157–164.

KING, V., RAO, S. and TARJAN, R. A Faster Deterministic Maximum Flow Algorithm. *Journal of Algorithms* 17 (1994), 447-474.

KOHLI, P., SHEKHOVTSOV, A., ROTHER, C., KOLMOGOROV, V. and TORR, P. *On partial optimality in multi-label MRFs*. Proceedings of the 25th international conference on Machine learning, ACM (2008), 480–487.

KOLMOGOROV, V. Convergent Tree-Reweighted Message Passing for Energy Minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28 (2006), 1568 -1583.

KOLMOGOROV, V. and ROTHER, C. Minimizing Nonsubmodular Functions with Graph Cuts-A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007), 1274 -1279.

KOLMOGOROV, V. and ZABIH, R. *Computing visual correspondence with occlusions using graph cuts*. Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings (2001), 508-515 vol.502.

KOLMOGOROV, V. and ZABIH, R. *Multi-camera Scene Reconstruction via Graph Cuts*, Springer Berlin / Heidelberg, (2002).

KOLMOGOROV, V. and ZABIH, R. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26 (2004), 147 -159.

KOMODAKIS, N., PARAGIOS, N. and TZIRITAS, G. MRF Energy Minimization and Beyond via Dual Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011), 531-552.

KOMODAKIS, N. and TZIRITAS, G. Approximate Labeling via Graph Cuts Based on Linear Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007), 1436 -1453.

KOVTUN, I. *Partial Optimal Labeling Search for a NP-Hard Subclass of (max,+) Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, (2003).

KUMAR, M. P. and KOLLER, D. *MAP estimation of semi-metric MRFs via hierarchical graph cuts*. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, AUAI Press (2009), 313–320.

KUMAR, M. P., VEKSLER, O. and TORR, P. H. S. Improved Moves for Truncated Convex Models. *J. Mach. Learn. Res.* 12 (2011), 31–67.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G. and BOBICK, A. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.* 22 (2003), 277–286.

LAFFERTY, J., MCCALLUM, A. and PEREIRA, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Departmental Papers (CIS)* (2001).

LEELHAPANTU, S. and CHALIDABHONGSE, T. H. *Stereo correspondence using multi-label QPBO method*. 2014 19th International Conference on Digital Signal Processing (DSP) (2014), 173-178.

LEELHAPANTU, S. and CHALIDABHONGSE, T. H. A speedup scheme for MRF stereo using local label hierarchy. *Signal, Image and Video Processing* (2018), 1-9.

LEELHAPANTU, S., KITTICHAROONWIT, P. and SURARERKS, A. *Generalization in Rational Base Number Representation Systems*. Proceedings of the 3th Regional Conference on ICT Application for Industries and Small Companies in ASEAN Countries (RCICT2011) (2011), 48-52.

LEMPITSKY, V., ROTHER, C. and BLAKE, A. *LogCut - Efficient Graph Cut Optimization for Markov Random Fields*. IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007 (2007), 1-8.

LEMPITSKY, V., ROTHER, C., ROTH, S. and BLAKE, A. Fusion Moves for Markov Random Field Optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32 (2010), 1392 -1405.

LI, M., SHEKHOVTSOV, A. and HUBER, D. *Complexity of Discrete Energy Minimization Problems*. Computer Vision – ECCV 2016, Springer, Cham (2016), 834-852.

LIU, K., ZHANG, J., YANG, P. and HUANG, K. *GRSA: Generalized range swap algorithm for the efficient optimization of MRFs*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015), 1761–1769.

LUO, W., SCHWING, A. G. and URTASUN, R. *Efficient Deep Learning for Stereo Matching*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), 5695-5703.

MAIRAL, J., ELAD, M. and SAPIRO, G. Sparse Representation for Color Image Restoration. *IEEE Transactions on Image Processing* 17 (2008), 53-69.

MEIR, O., GALUN, M., YAGEV, S., BASRI, R. and YAVNEH, I. *A Multiscale Variable-Grouping Framework for MRF Energy Minimization*. 2015 IEEE International Conference on Computer Vision (ICCV) (2015), 1805–1813.

METSIRITRAKUL, K., PUNTAVACHIRAPAN, N., KOBCHAISAWAT, T., LEELHAPANTU, S. and CHALIDABHONGSE, T. H. *UP2U: Program for raising awareness of phubbing problem with stimulating social interaction in public using augmented reality and computer vision*. 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) (2016), 1-6.

NIEUWENHUIS, C., TÖPPE, E. and CREMERS, D. A Survey and Comparison of Discrete and Continuous Multi-label Optimization Approaches for the Potts Model. *International Journal of Computer Vision* (2013), 1-18.

PHILLIPS, S. and WESTBROOK, J. *Online Load Balancing and Network Flow*. Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing, ACM (1993), 402–411.

PLETSCHER, P., NOWOZIN, S., KOHLI, P. and ROTHER, C. *Putting MAP Back on the Map*, Springer Berlin Heidelberg, Berlin, Heidelberg, (2011).

PRASONGPONGCHAI, T., CHALIDABHONGSE, T. H. and LEELHAPANTU, S. *A vision-based method for the detection of missing rail fasteners*. 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) (2017), 419-424.

ROTHER, C., KOLMOGOROV, V., LEMPITSKY, V. and SZUMMER, M. *Optimizing binary MRFs via extended roof duality*. Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (2007), 1–8.

ROTHER, C., KUMAR, S., KOLMOGOROV, V. and BLAKE, A. *Digital tapestry [automatic image synthesis]*. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (2005), 589–596.

ROY, S. Stereo Without Epipolar Lines: A Maximum-Flow Formulation. *International Journal of Computer Vision* 34 (1999), 147-161.

RUEOPAS, W., LEELHAPANTU, S. and CHALIDABHONGSE, T. H. *A corner-based saliency model*. 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE) (2016), 1-6.

SAVCHYNSKYY, B., KAPPES, J. H., SWOBODA, P. and SCHNÖRR, C. *Global MAP-Optimality by Shrinking the Combinatorial Search Area with Convex Relaxation*, Curran Associates, Inc., (2013).

SCHARSTEIN, D., HIRSCHMÜLLER, H., KITAJIMA, Y., KRATHWOHL, G., NEŠIĆ, N., WANG, X. and WESTLING, P. *High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth*, Springer International Publishing, (2014).

SCHARSTEIN, D. and PAL, C. *Learning Conditional Random Fields for Stereo*. Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on (2007), 1-8.

SCHARSTEIN, D. and SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision* 47 (2002), 7–42.

SCHARSTEIN, D. and SZELISKI, R. *High-accuracy stereo depth maps using structured light*. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings (2003), I-195 - I-202 vol.191.

SCHLESINGER, D. *Exact Solution of Permuted Submodular MinSum Problems*, Springer Berlin Heidelberg, (2007).

SCHLESINGER, D. and FLACH, B. *Transforming an arbitrary minsum problem into a binary one*, TU, Fak. Informatik, (2006).

SHEKHOVTSOV, A., KOLMOGOROV, V., KOHLI, P., HLAVÁC, V., ROTHER, C. and TORR, P. *LP-relaxation of binarized energy minimization*. Research Report CTU–CMP–2007–27). Czech Technical University (2008).

SLEATOR, D. D. and ENDRE TARJAN, R. A data structure for dynamic trees. *Journal of Computer and System Sciences* 26 (1983), 362-391.

SZELISKI, R., ZABIH, R., SCHARSTEIN, D., VEKSLER, O., KOLMOGOROV, V., AGARWALA, A., TAPPEN, M. and ROTHER, C. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30 (2008), 1068 - 1080.

TANIAI, T., MATSUSHITA, Y. and NAEMURA, T. *Graph Cut Based Continuous Stereo Matching Using Locally Shared Labels*. 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014), 1613–1620.

TAPPEN, M. F. and FREEMAN, W. T. *Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters*. Ninth IEEE International Conference on Computer Vision, 2003. Proceedings (2003), 900-906 vol.902.

TARLOW, D., GIVONI, I. E., ZEMEL, R. S. and FREY, B. J. *Graph Cuts is a Max-Product Algorithm.* UAI (2011), 671–680.

THOMO, I., MALASIOTIS, S. and STRINTZIS, M. G. *Optimized block based disparity estimation in stereo systems using a maximum-flow approach*. International Symposium on Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98 (1998), 410-417.

TIEFENBACHER, P., BOGISCHEF, V., MERGET, D. and RIGOLL, G. *Subjective and objective evaluation of image inpainting quality*. 2015 IEEE International Conference on Image Processing (ICIP) (2015), 447-451.

VEKSLER, O. *Reducing Search Space for Stereo Correspondence with Graph Cuts.* BMVC (2006), 709–718.

VEKSLER, O. Multi-label Moves for MRFs with Truncated Convex Priors. *International Journal of Computer Vision* 98 (2012), 1-14.

WAINWRIGHT, M. J., JAAKKOLA, T. S. and WILLSKY, A. S. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory* 51 (2005), 3697-3717.

WEISS, Y. and FREEMAN, W. T. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Computation* 13 (2001), 2173–2200.

WOODFORD, O., TORR, P., REID, I. and FITZGIBBON, A. Global Stereo Reconstruction under Second-Order Smoothness Priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (2009), 2115 -2128.

XIAO, M. *Algorithms for Multiterminal Cuts*, Springer Berlin Heidelberg, (2008).

XUE, H., ZHANG, S. and CAI, D. Depth Image Inpainting: Improving Low Rank Matrix Completion With Low Gradient Regularization. *IEEE Transactions on Image Processing* 26 (2017), 4311-4320.

YANOVER, C., SCHUELER-FURMAN, O. and WEISS, Y. Minimizing and Learning Energy Functions for Side-Chain Prediction. *Journal of Computational Biology* 15 (2008), 899-911.

YEDIDIA, J. S., FREEMAN, W. T. and WEISS, Y. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium* 8 (2003), 236–239.

YEH, R. A., CHEN, C., LIM, T. Y., SCHWING, A. G., HASEGAWA-JOHNSON, M. and DO, M. N. *Semantic Image Inpainting with Deep Generative Models*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017), 6882-6890.

ZACH, C. *A Principled Approach for Coarse-to-Fine MAP Inference*. 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014), 1330–1337.

ZHANG, K., ZUO, W., CHEN, Y., MENG, D. and ZHANG, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing* 26, 7 (2017), 3142-3155.

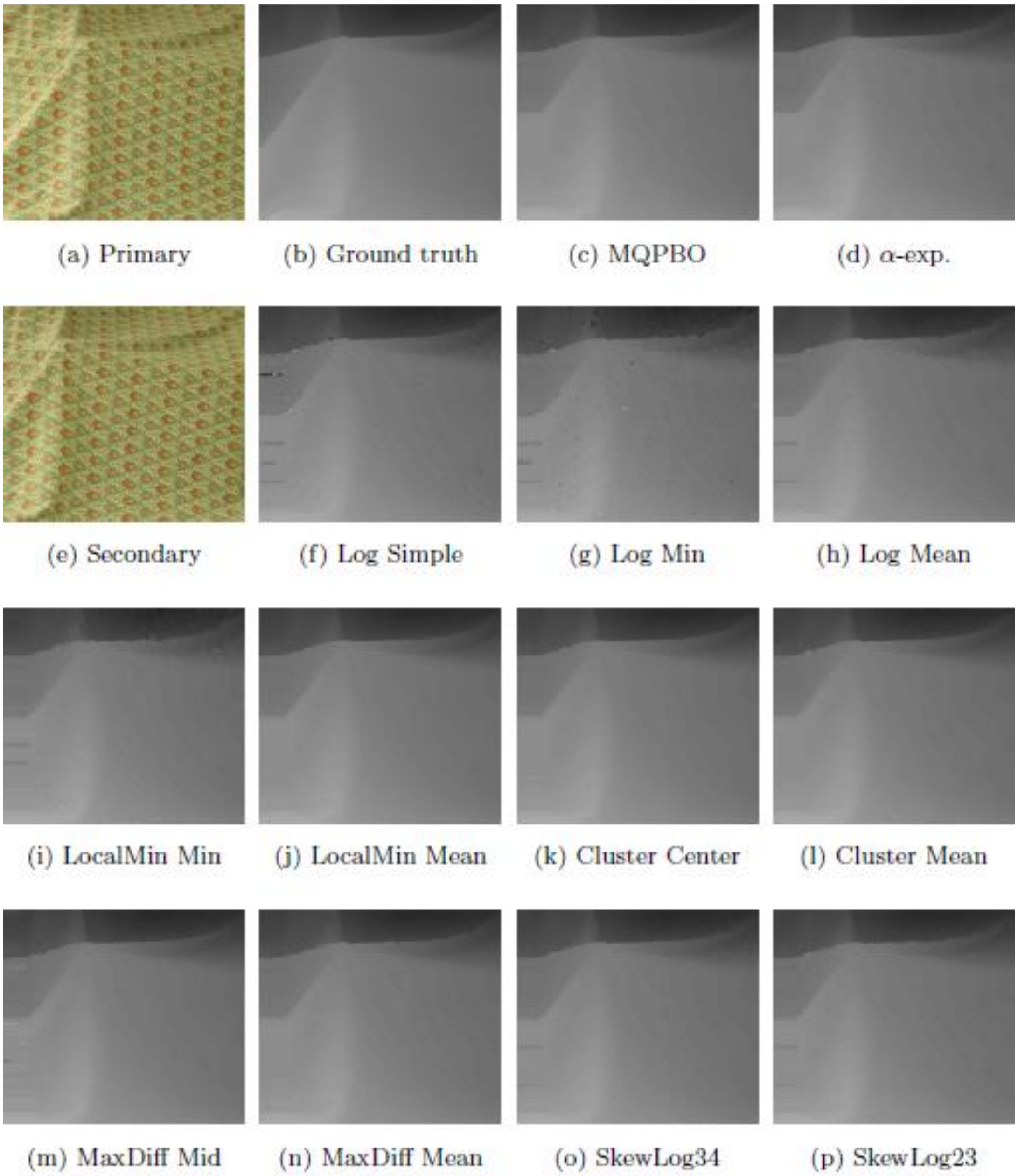
ZHANG, Y., HARTLEY, R. and WANG, L. *Fast Multi-labelling for Stereo Matching*. Computer Vision – ECCV 2010, Springer, Berlin, Heidelberg (2010), 524–537.

ZHAO, H. *Global optimal surface from stereo*. 15th International Conference on Pattern Recognition, 2000. Proceedings (2000), 101-104 vol.101.

APPENDIX

Appendix A: Additional Results

Figure 24, Figure 25, Figure 26, and Figure 27 show the input images, ground truth images, and the disparity maps of the stereo correspondence solutions computed by the techniques being compared using sum of square differences and Gaussian smoothness prior. Overall, the techniques exhibit the same relative qualitative results as discussed in section 4.2.



(a) Primary (b) Ground truth (c) MQPBO (d) α-exp.

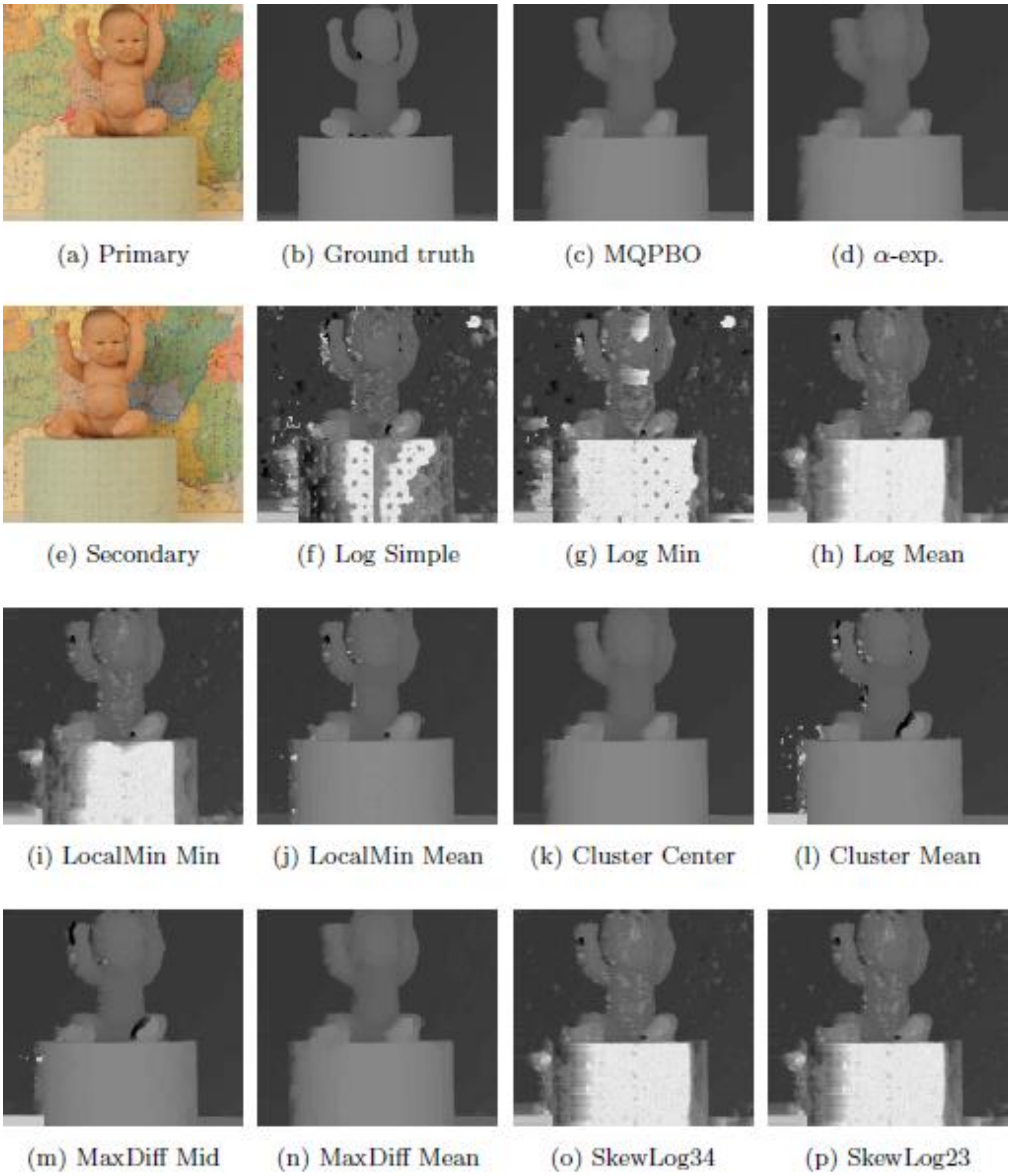
(e) Secondary (f) Log Simple (g) Log Min (h) Log Mean

(i) LocalMin Min (j) LocalMin Mean (k) Cluster Center (l) Cluster Mean

(m) MaxDiff Mid (n) MaxDiff Mean (o) SkewLog34 (p) SkewLog23

Figure 24: Disparity maps of the Cloth1 stereo pair

For the Cloth1 stereo pair shown in Figure 24, the results overall appear smoother than other pairs since the input images contain virtually no occluded part nor discontinuity. Artifacts can be seen along the left edge of the results where there are no pixels in the secondary image to match. Scattered artifacts are most obviously in **Log Min**.


(a) Primary (b) Ground truth (c) MQPBO (d) α-exp.

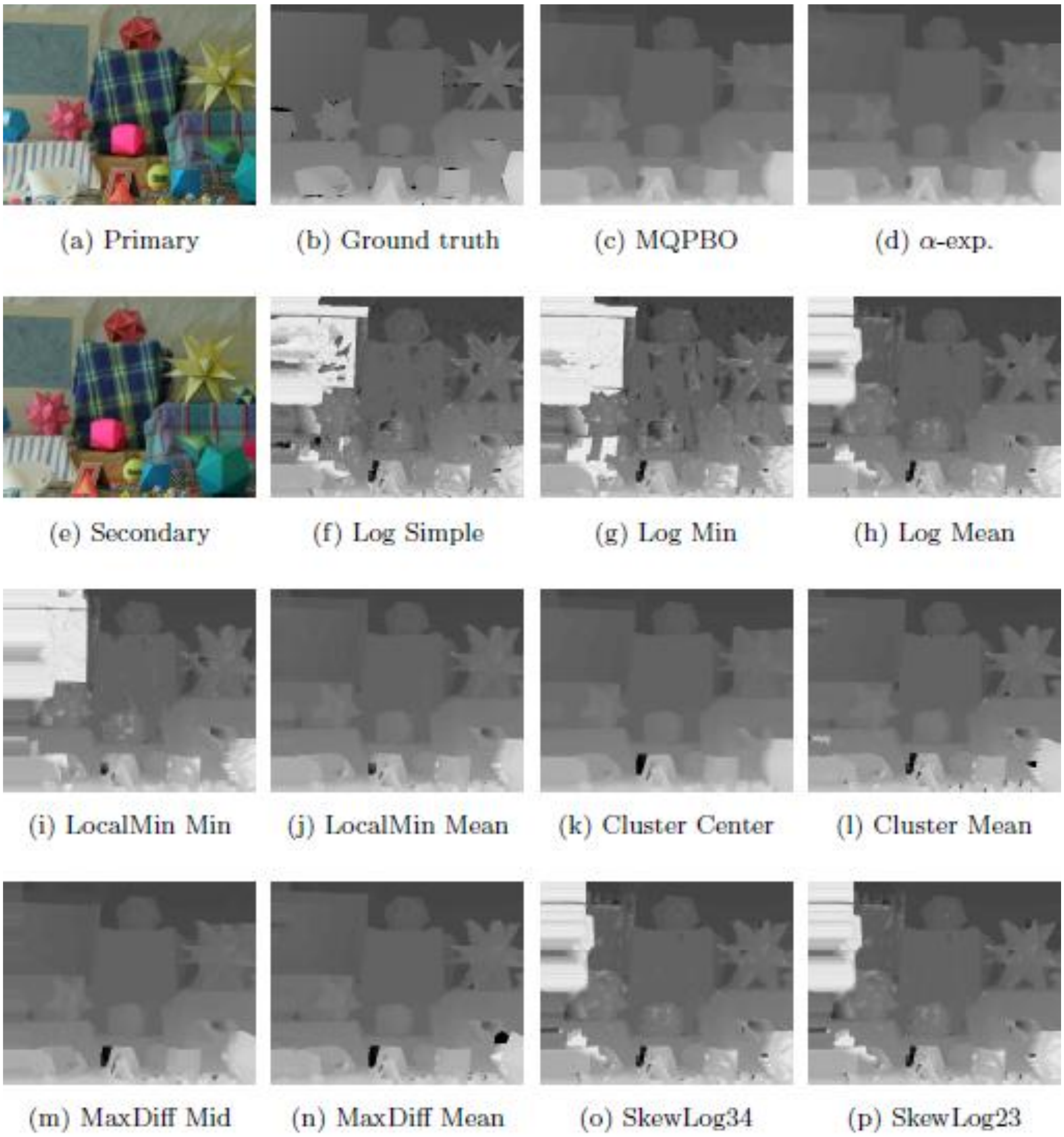
(e) Secondary (f) Log Simple (g) Log Min (h) Log Mean

(i) LocalMin Min (j) LocalMin Mean (k) Cluster Center (l) Cluster Mean

(m) MaxDiff Mid (n) MaxDiff Mean (o) SkewLog34 (p) SkewLog23

Figure 25: Disparity maps of the Baby1 stereo pair

For the Baby1 stereo pair (Figure 25), the **Log** and **SkewLog** techniques give unfavorable results. The region in question has disparity values in the range near 128, which is the first splitting point of binary subdivision. **LocalMin** techniques also happen to divide the group around this range and the resulting disparity maps, again, show how using **Mean** is better than **Min** for **LocalMin**.



(a) Primary (b) Ground truth (c) MQPBO (d) α-exp.

(e) Secondary (f) Log Simple (g) Log Min (h) Log Mean

(i) LocalMin Min (j) LocalMin Mean (k) Cluster Center (l) Cluster Mean

(m) MaxDiff Mid (n) MaxDiff Mean (o) SkewLog34 (p) SkewLog23

Figure 26: Disparity maps of the Moebius stereo pair

Figure 26 shows the Moebius stereo pair. Like Cloth1, most artifacts are along the left edge since there are no pixels in the secondary image to match. Unlike Cloth1, however, there are several occluded regions in this pair. The most noticeable is the region to the left of the cards where the results from grouping techniques return disparity value of zero but the baseline techniques do not.

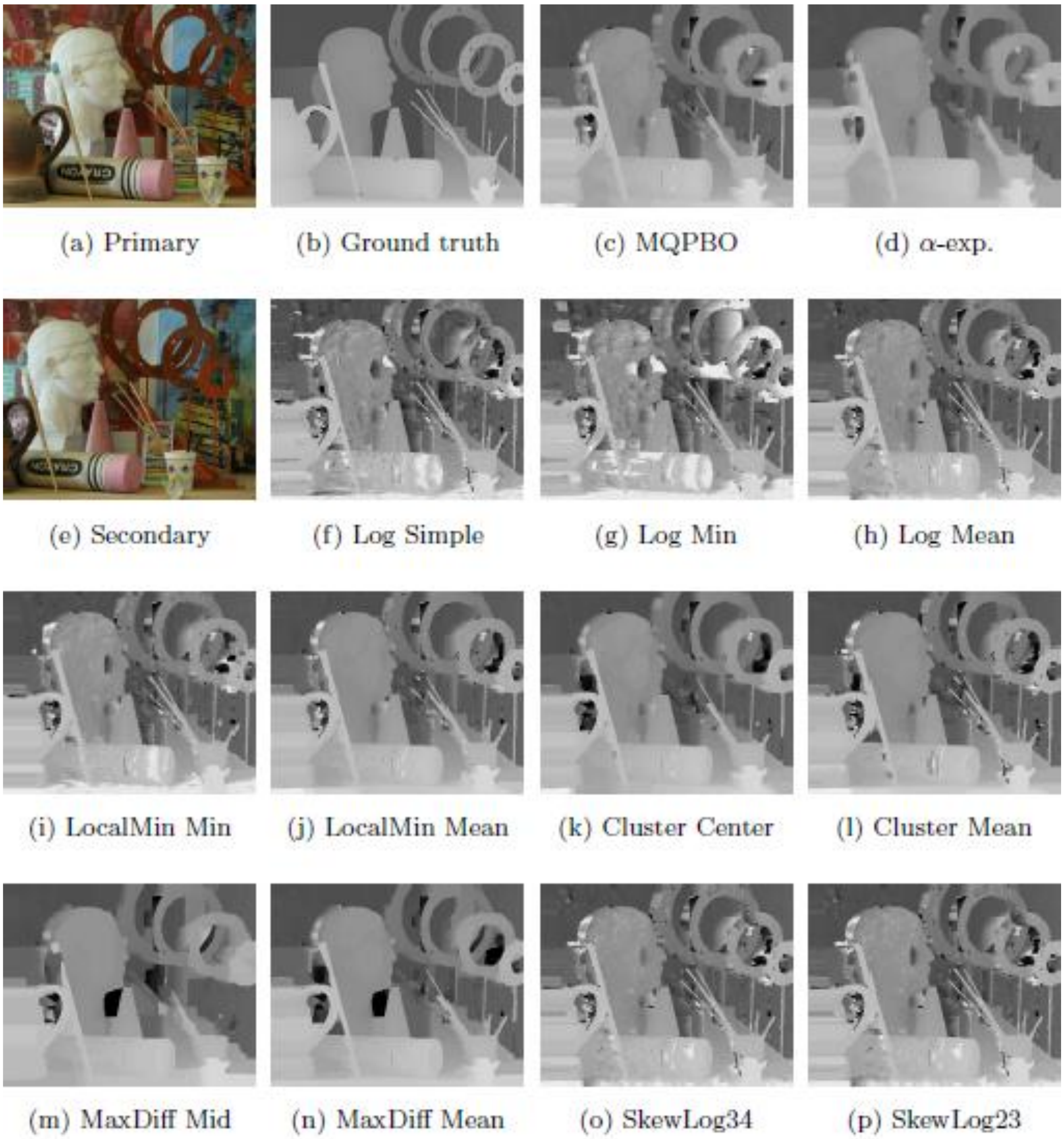
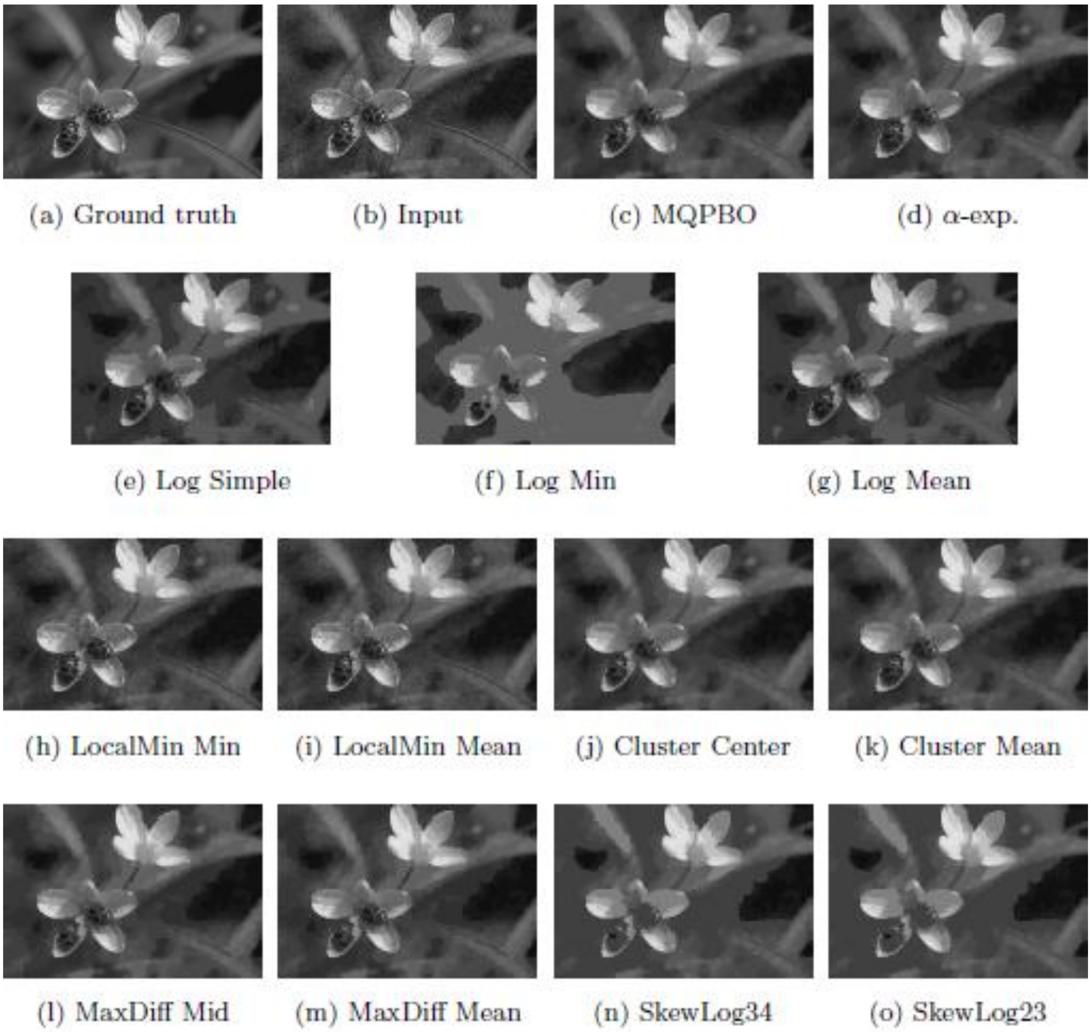


Figure 27: Disparity maps of the Art stereo pair

Figure 27 shows the Art stereo pair. The slender objects on the right are missing due to over-smoothing in the baseline α-expansion techniques and some grouping techniques. The numbers of error pixels in these cases, however, are still lower than those of **Log** and **SkewLog** techniques since these objects occupy a very small portion of the scene and the error elsewhere outweighs them.


(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

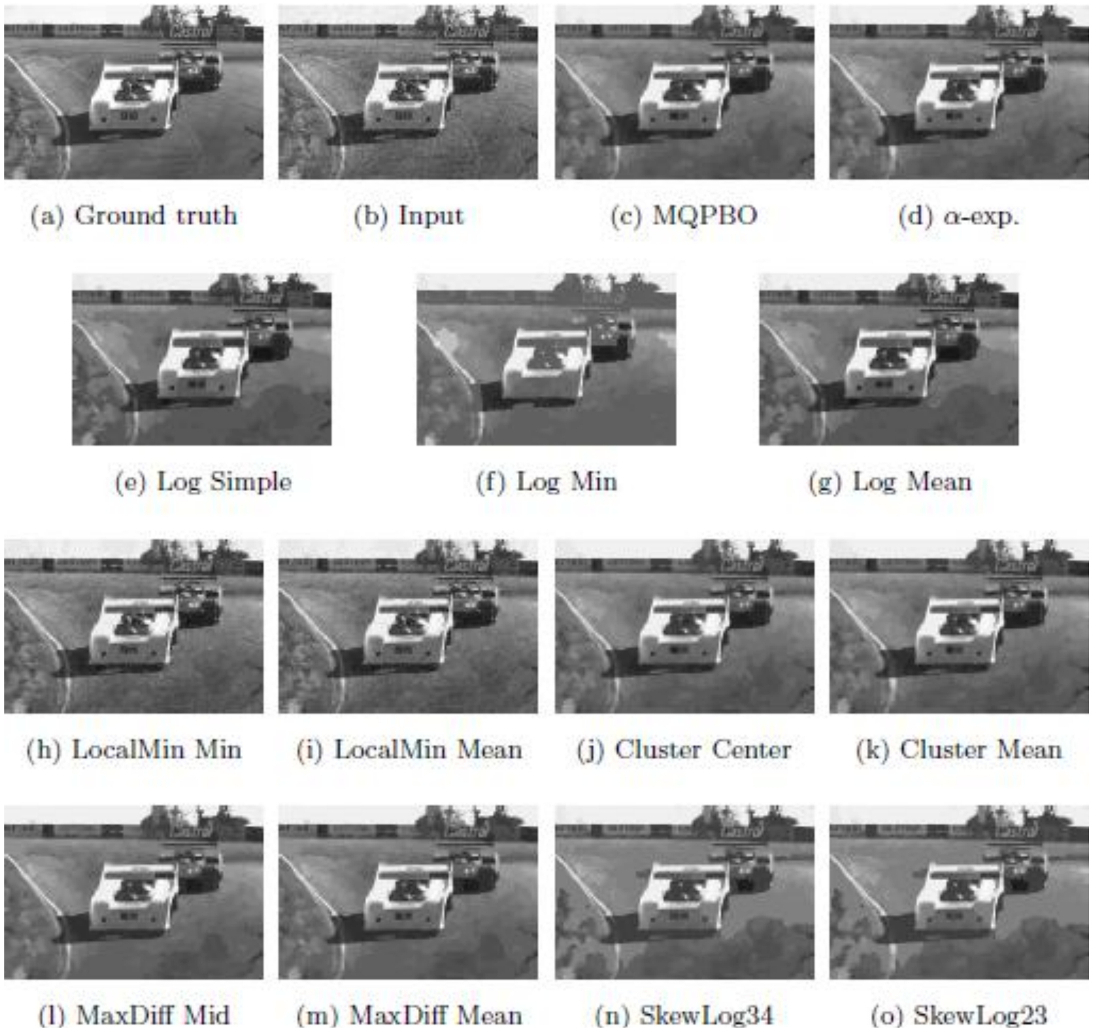
(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 28: Denoising results of "Flowers" image

Figure 28 and Figure 29 show the input images, ground truth images, and the denoising results computed by the techniques being compared from using sum of square-root of absolute differences cost and inverse smoothness prior. Table 10 shows their PSNR results.

In the "Flowers" dataset, the artifacts from over-smoothing are more pronounced for **Log** and **SkewLog** techniques with **Log Min** being most noticeable. The tradeoff between preserving and smoothing pixel intensity values can be observed by comparing **LocalMin Mean** and **MaxDiff Mean** as **MaxDiff** over-smoothed the middle-right region which results in the obtrusive boundary between the darker and brighter area.

Comparing the PSNR in Table 10, however, reveals that preserving noise in **LocalMin** gives worse quantitative results.



(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 29: Denoising results of "Race cars" image

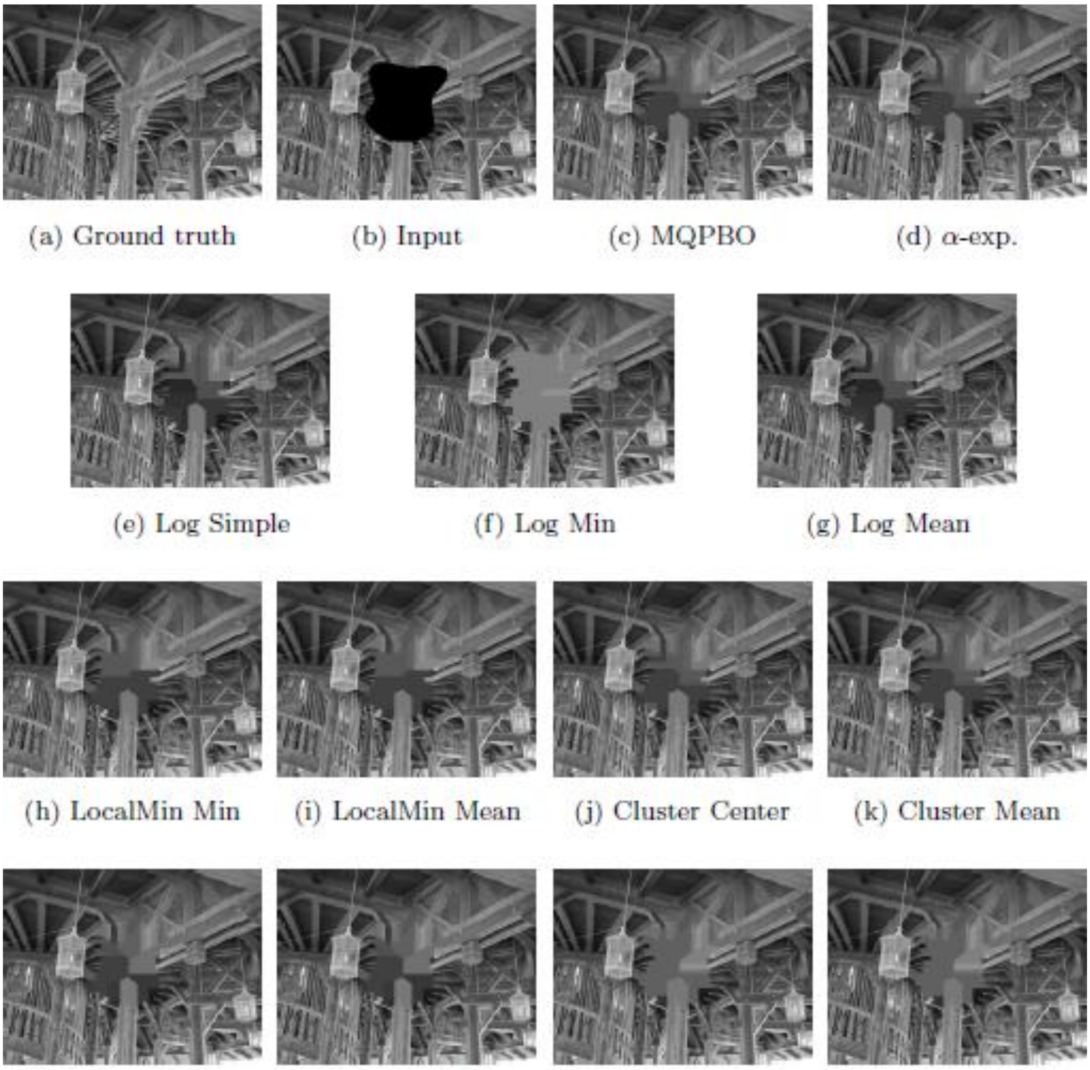
The artifacts in the "Race cars" dataset are most perceptible at the car in the center and on the bottom-right darkened area. As with before, smoothing the noisy pixels can lead to unwanted observable boundaries between regions but preserving the noise leads to worse PSNR results.

Table 10: PSNR results of "Flowers" and "Race cars" datasets

Flowers Techniques	PSNR	Race cars Techniques	PSNR
Input	26.955	Input	26.8973
MQPBO	30.767	MQPBO	28.6547
AlphaExp	30.785	AlphaExp	28.6613
Log Simple	28.692	Log Simple	27.6551
Log Min	27.188	Log Min	27.8455
Log Mean	28.726	Log Mean	27.6632
LocalMin Min	28.988	LocalMin Min	27.7008
LocalMin Mean	28.997	LocalMin Mean	27.7098
Cluster Center	30.685	Cluster Center	28.619
Cluster Mean	30.675	Cluster Mean	28.619
MaxDiff16 Mid	30.684	MaxDiff16 Mid	28.6412
MaxDiff16 Mean	30.685	MaxDiff16 Mean	28.6423
SkewLog 3/4	28.709	SkewLog 3/4	27.7183
SkewLog 2/3	28.709	SkewLog 2/3	27.7134

Figure 30 and Figure 31 show the ground truth images, input images, and the inpainting results computed by the techniques being compared from using sum of square-root of absolute differences cost and inverse smoothness prior. The corresponding PSNR values are given in Table 11. The values for **Cluster** techniques are from the delayed version (section 4.4).

Due to having a rather large unknown region in the "12" dataset, the inpainting model can only project the structures of the scene to a certain extent (with no texture). Comparing **Log Simple** and **Log Min** with the **SkewLog** techniques, the results from the **Log** techniques appear more pleasing to the eye while the results from the **SkewLog** techniques gives higher PSNR values.



(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

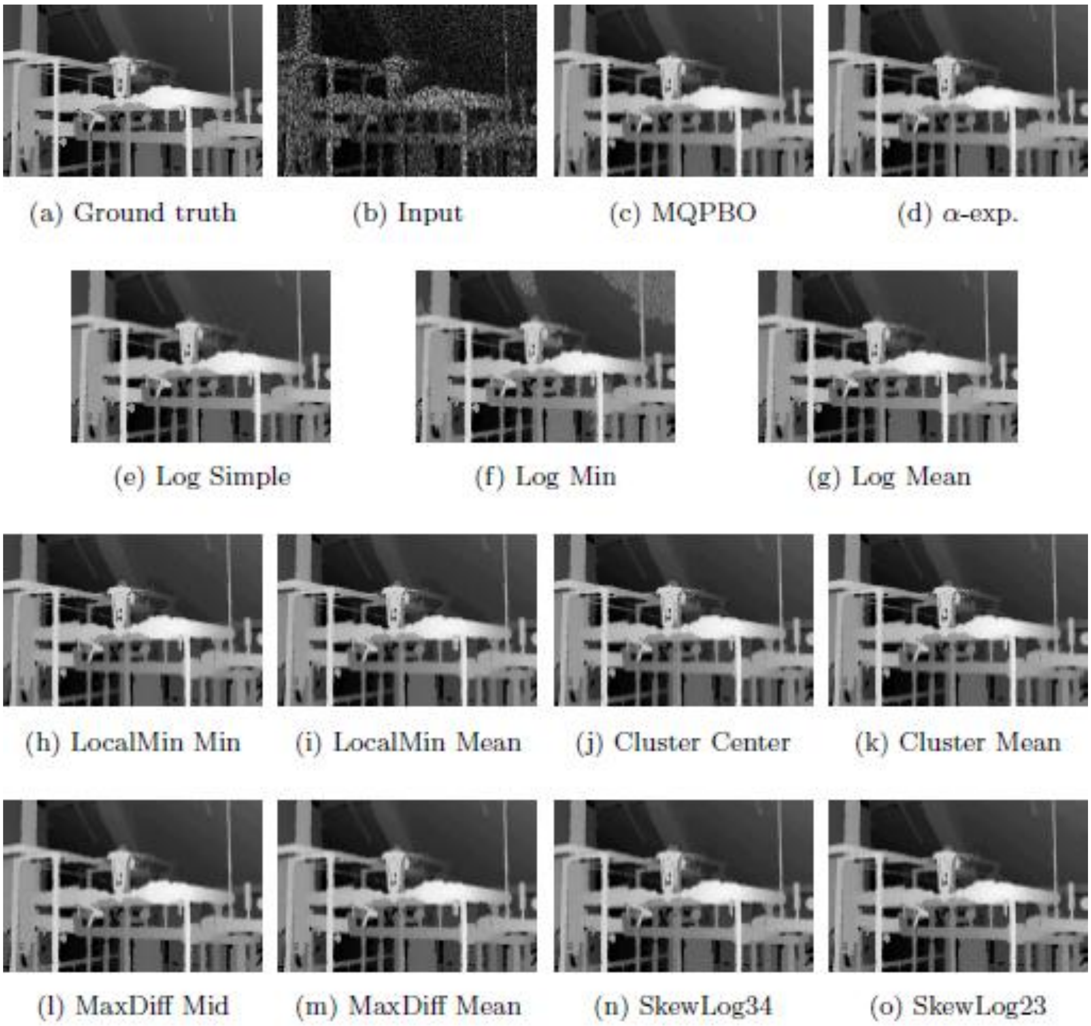
(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 30: Inpainting results of "12" from the TUM-IID dataset

As with other datasets, the most noticeable artifacts in the "Pipes" dataset appear in the result from **Log Min** technique. In the rest of the techniques, most artifacts can be seen along the edges of the pipes, where the depth discontinuities can lead the model to make wrong decisions.



(a) Ground truth (b) Input (c) MQPBO (d) α-exp.

(e) Log Simple (f) Log Min (g) Log Mean

(h) LocalMin Min (i) LocalMin Mean (j) Cluster Center (k) Cluster Mean

(l) MaxDiff Mid (m) MaxDiff Mean (n) SkewLog34 (o) SkewLog23

Figure 31: Inpainting results of "Pipes" from the Depth Inpainting database

Table 11: PSNR results of "12" and "Pipes" datasets

12 **Techniques**	**PSNR**		**Pipes** **Techniques**	**PSNR**
MQPBO	25.729		MQPBO	32.132
AlphaExp	25.73		AlphaExp	32.134
Log Simple	25.431		Log Simple	29.651
Log Min	25.475		Log Min	29.012
Log Mean	25.442		Log Mean	29.652

12 **Techniques**	**PSNR**
LocalMin Min	25.452
LocalMin Mean	25.451
Cluster Center	25.73
Cluster Mean	25.73
MaxDiff16 Mid	25.728
MaxDiff16 Mean	25.729
SkewLog 3/4	25.691
SkewLog 2/3	25.69

Pipes **Techniques**	**PSNR**
LocalMin Min	31.743
LocalMin Mean	31.743
Cluster Center	32.114
Cluster Mean	32.089
MaxDiff16 Mid	32.092
MaxDiff16 Mean	32.099
SkewLog 3/4	30.877
SkewLog 2/3	30.617

Also note that even though only 10% of the pixels in the "12" dataset are unknown whereas the unknown pixels occupy more than 50% of the "Pipes" dataset, both qualitative and quantitative results of "Pipes" are better than those of "12". This is the typical behavior of inpainting algorithms since the unknown pixels are inferred from the known pixels.

Appendix B: Publication List

Related publications:

- [Leelhapantu and Chalidabhongse 2018]
- [Leelhapantu and Chalidabhongse 2014]

Other publications:

- [Prasongpongchai et al. 2017]
- [Rueopas et al. 2016]
- [Metsiritrakul et al. 2016]
- [Leelhapantu et al. 2011]

VITA

Sangsan Leelhapantu grew up in Bangkok, Thailand, eventually studying at Sirirattanathorn School before he attended Chulalongkorn University from 2007 to 2011, and graduated 1st class honors, with a B.Eng. degree in Computer Engineering. While there he performed senior project with Asst. Prof. Dr. Athasit Surarerks on generalization in rational base number representation systems.

After undergraduate school, he entered the Ph.D. program at Chulalongkorn University under Chulalongkorn University graduate scholarship to commemorate the 72nd anniversary of His Majesty King Bhumibol Adulyadej. The author became interested in computer vision and probabilistic graphical models, which he continued to research under the advisorship of Asst. Prof. Dr. Thanarat Chalidabhongse.