

Stereo Matching Using Higher-Order Graph Cuts

Yiran Xie

June 2012

A thesis submitted for the degree of Master of Philosophy
of the Australian National University



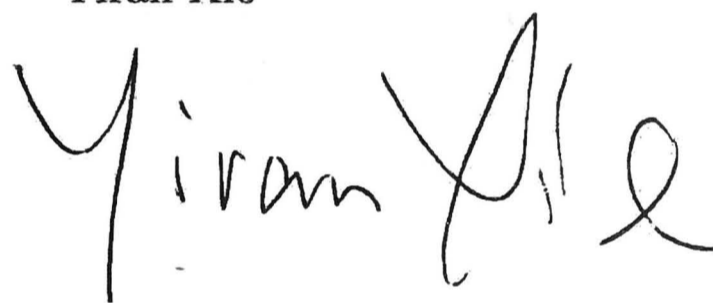
**Australian
National
University**

To my parents

Declaration

The work in this thesis is my own except where otherwise stated.

Yiran Xie

A handwritten signature in black ink that reads "Yiran Xie". The signature is written in a cursive style with a large, sweeping initial 'Y' and a long, horizontal tail on the 'e'.

Acknowledgements

This thesis would not have been done without the help and encouragement of a number of people. First, I would like to express my most profound and sincere thanks to my principal supervisor Nianjun Liu, who made it possible for me to come to ANU to pursue my Mphil study. During my stay, I am not only taught in academic skills but also received generous support. Second, I would like to thank my co-supervisor Xuming He for the many enlightening discussions we had. Third, many thanks to all members in my previous or current supervisor panel (Nick Barnes, Hongdong Li and Lei Wang), I also benefit a lot from them. Fourth, I am indebted to Sheng Liu who was a visiting scholar, I appreciate the one year time we spent together, and his passion for research is infectious.

During my study, I am fortunate to participate in the VIBE project, I would like to thank Paulette Lieby who gave me this opportunity, and all my colleagues for their collaboration.

My stay in the NICTA Canberra lab was made pleasurable by numerous friends and colleagues who I would like to thank for their company, especially Lin Gu, Jun Sun, Tao Wang, Hanxi Li. I will never forget the trips we went together, the dinners we had and the foosball games we played, life would be dim without them.

Outside my academic life, I am grateful to all my friends, I will always keep the beautiful memories. Forgive me not have listed all your names here, but I sincerely appreciate all your companionship.

Abstract

Stereo matching is one of the fundamental tasks in early vision. Unlike human brain recognizes objects and estimates the depth easily, it is difficult to design algorithms that perform well on a computer due to variations of illumination, occlusion or textureless. Like most of the early vision problems, stereo matching can be formulated as an energy minimization problem in which the optimal depth is the one with the lowest energy. And graph cuts is one of the efficient and effective minimization tools that avoids the problems of local minima. Conventional energy functions are defined on Markov Random Fields(MRFs) with a 4-connected grid structure derived from the image, however it is incapable of expressing complex relationship between group of pixels. This thesis focuses on exploring some aspects of stereo matching problems through higher-order structure and higher-order graph cuts.

The first problem I address relates to the evaluation of five state-of-the-art segmentation approaches. Their different contributions to segment-based stereo matching have been quantitatively measured and analyzed. This works aim at helping researchers to choose the segmentation approach that most suitable for their stereo matching application.

The second part of the thesis proposes a novel approach to dense stereo matching. This method features sub-segmentation and adopts a higher-order potential to enforce the label consistency inside segments as a soft constraint. Moreover, several successful techniques have been combined. Experiments show that this approach obtains state-of-the-art results while still keeping efficiency.

In the last part of the thesis, a novel two-layer MRFs framework is presented in which stereo matching and surface boundary estimation are combined. Both properties are inferred simultaneously and globally so that they can benefit each other. This work has direct application in phosphene vision based human indoor navigation. Experiments prove that the proposed framework achieves significantly better performance than other popular methods in all resolutions.

Publications

Several contributions in this thesis have been published elsewhere. We list these below:

- Yiran Xie, Nianjun Liu, Nick Barnes, ‘Phosphene Vision of Depth and Boundary from Segmentation-based Associative MRFs.’ In *Proceedings of 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society(EMBC 2012)*, San Diego, USA, 2012
- Yiran Xie, Nianjun Liu, Xuming He, Nick Barnes, ‘Joint Optimization on Coupled MRFS For Stereo Matching And Boundary Estimation.’ Submitted to *Proceedings of International Conference on Image Processing(ICIP 2012)*, Florida, USA, 2012
- Yiran Xie, Nianjun Liu, Sheng Liu, Nick Barnes, ‘Stereo Matching Using Sub-segmentation and Robust Higher-order Graph Cut.’ In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications(DICTA 2011)*, Queensland, Australia, 2011
- Yiran Xie, Rui Cao, Hangyang Tong, Sheng Liu, Nianjun Liu, ‘Evaluating Multi-scale Over-segment and Its Contribution to Real Scene Stereo Matching by High-Order MRFs.’ In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications(DICTA 2010)*, Sydney, Australia, 2010

Contents

Acknowledgements	vii
Abstract	ix
Publications	xi
1 Introduction	1
1.1 Computer Vision	1
1.2 Stereo Matching	1
1.3 Outline of the Thesis	3
2 Theoretical Backgrounds and Related Work	5
2.1 Bayesian Labeling and Markov Random Field	5
2.1.1 Labeling Problems	5
2.1.2 Markov Random Field	7
2.2 Inference	9
2.2.1 Iterated Conditional Models(ICM)	10
2.2.2 Graph Cut	10
2.2.3 Loopy Belief Propagation(LBP)	11
2.2.4 Dynamic Programming(DP)	12
2.2.5 Tree-Reweighted Message Passing(TRW)	12
2.3 Pseudo-boolean Optimization and Graph Cuts	13
2.3.1 Graph Cuts in Computer Vision	13
2.3.2 Pseudo-boolean Representation	14
2.3.3 Energy Cost and Graph Representation	16
2.3.4 Max-Flow Algorithms	20
2.4 Stereo Matching	23
2.4.1 The Two-Frame Stereo Matching Problem	23
2.4.2 Matching Constraints	25

2.4.3	Pixel-Based and Segment-Based Algorithms	26
2.4.4	Local and Global Algorithms	27
3	Evaluation of Different Segmentation Algorithms and Their Performance in Stereo Matching	31
3.1	Overview of Image Segmentation and its Evaluation	31
3.2	Five Modern Segmentation Algorithms	32
3.3	Segmentation Evaluation Framework	34
3.3.1	Analytical Methods	34
3.3.2	Empirical Discrepancy Methods	35
3.3.3	Empirical Goodness Methods	35
3.4	Evaluation of Performance in Stereo Matching	37
3.5	Experiment	40
3.5.1	Empirical Goodness Evaluation	40
3.5.2	Performance Evaluation in Segment-based Stereo Matching	42
4	Stereo Matching Using Sub-segmentation and Robust Higher-order Graph Cuts	49
4.1	Introduction	49
4.2	Stereo Matching Through Robust Higher-order Graph Cuts	51
4.2.1	Initial Steps	51
4.2.2	Occlusion Handling	52
4.2.3	Confidence Measurement	55
4.2.4	Plane Fitting	56
4.2.5	Sub-segmentation	59
4.2.6	Energy Function Model	61
4.2.7	Robust Higher-Order Term and Graph Cuts	61
4.3	Experiment	63
4.3.1	Quality	63
4.3.2	Efficiency and Energy Convergence Analysis	65
5	Joint Optimization on Coupled MRFs for Stereo Matching and Surface Boundary Estimation	69
5.1	Motivation and Introduction	69
5.2	Triangulation-Based Joint Framework for Stereo and Surface Boundary Completion	72
5.2.1	Boundary Potentials	73
5.2.2	Surface Boundary Potentials	74

5.2.3	Stereo Matching Potentials	75
5.2.4	Interaction Potentials	76
5.2.5	Joint Inference	77
5.3	Segment-Based Joint Framework for Phosphene Vision in Indoor Navigation	77
5.3.1	Downsampling and Phosphene Representation	79
5.4	Experiment	80
5.4.1	Experiment of Triangulation-Based Algorithm in Surface Completion	81
5.4.2	Experiment of Segment-Based Algorithm in Human Navi- gation	81
6	Conclusions	87
6.1	Contributions	87
6.2	Future Works	88
6.2.1	Objects Recognition	88
6.2.2	Hierarchical Model in Stereo Matching	89
6.2.3	Projection Graph Cuts for Problems with Large Label Space	89
	Bibliography	91

Chapter 1

Introduction

1.1 Computer Vision

Like human using their eyes to perceive the real world, the theme of computer vision is to simulate the human vision by machines to analyze and understand images or video sequences. As a joint discipline, computer vision closely relates to the fields of physics, signal processing, artificial intelligence and machine learning. The attention of computer vision has been well paid since the 1970s along the development of computational abilities and maturing of active applications: autonomous vehicle navigation, medical imaging, automatic surveillance and others[68].

Generally, computer vision consists of three levels of tasks: low-level, middle-level and high-level. Low-level computer vision, or commonly known as early vision, is mainly confronted with the tasks of acquiring features and recovering three-dimensional shape from images. Middle-level and high-level vision problems, on the other hand, focus on object detection and scene understanding. According to[57], early vision tasks are usually “inverse” problems, thus they are ill-posed. Their solutions are not unique, and often problems themselves are not sufficiently constrained. To regularize them, researchers have to introduce specific constraints to the problems. Typical early vision tasks include stereo matching, image restoration and image segmentation.

1.2 Stereo Matching

Stereo matching is one of the most heavily researched topics in early vision. It has a wide range of potential applications including: three-dimensional scene-

reconstruction, robot navigation. Comprehensive reviews can be found in [67][5][12][18]. In general, it can be sorted into two categories: two-frame and multi-frame. In the two-frame stereo matching problem, a scene is captured by two cameras simultaneously. The main purpose is for computers to predict the distance between the objects and the cameras through these two captured images. If these two images have already been well rectified, the positions of an object will have a horizontal shift in two images depending on its distance to the camera. This distance is usually called disparity, and is inversely proportional to the depth. Therefore the main process of stereo matching is to find the correspondence between the two images and compute their disparity. The disparity information of each pixel can be displayed in the form of a disparity map(as shown in Figure 1.1), in which the brighter the pixel is, the closer it is to the camera.

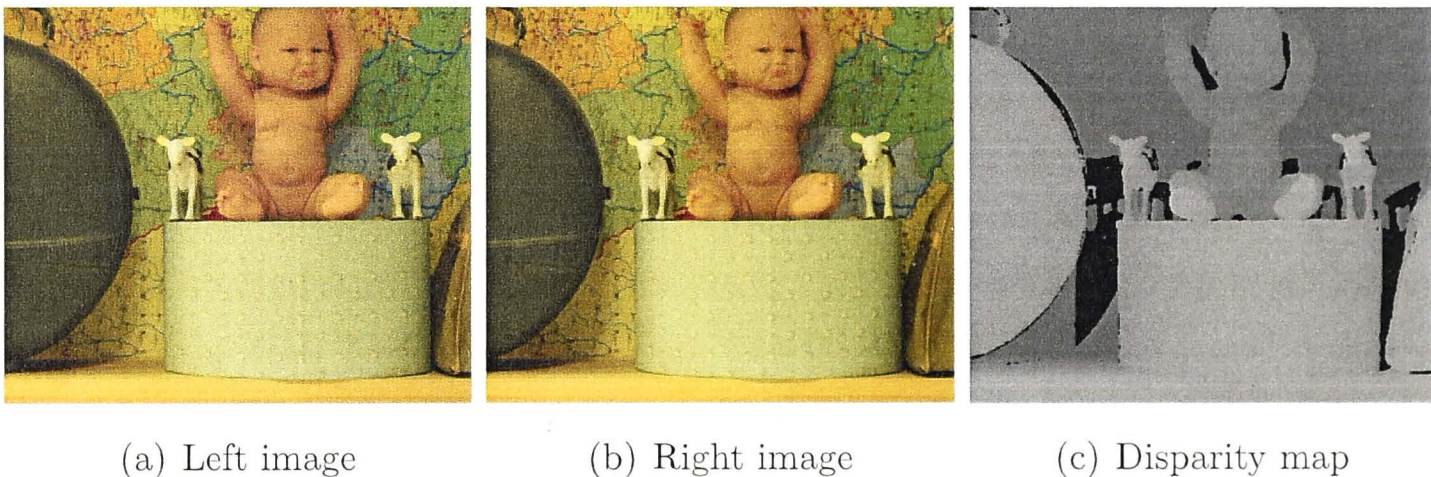


Figure 1.1: An example from Middlebury[67].

A variety of constraints are used to guide the correspondence solution including photoconsistency, continuity, uniqueness, ordering and others. Algorithms for stereo matching can be divided into local approaches and global approaches. Local approaches neglect the smoothness of spatially neighboring pixels and usually their solutions are decided by a pixel independent “winner-take-all” strategy. In practice, local approaches are efficient but not robust enough. Meanwhile global approach formulate itself as a pre-defined energy minimization problem in which the lowest energy corresponds to the optimal labeling. In their energy functions, they usually have smoothness terms penalizing on difference between neighboring pixels. In such design, pixels are optimized in the same global framework. Most of the existing global algorithms are defined on Markov Random Fields(MRFs) with a 4-connected grid structure, however it is incapable of expressing complex relationship between group of pixels. In this thesis, we focus on exploring some of the higher-order structure of stereo matching.

1.3 Outline of the Thesis

A brief outline of the thesis follows.

1. In chapter 2, we first review the basic concepts used in this thesis including Markov Random Field, inference and graph cuts. Then we discuss the two-frame stereo matching problems and the limitations of previous work.
2. In chapter 3, we introduce the evaluation framework of five state-of-the-art segmentation approaches. In addition, their different contributions to segment-based stereo matching have been quantitatively measured and analyzed.
3. In chapter 4, we propose a new approach to dense stereo matching. It features sub-segmentation and adopts a higher-order potential to enforce the label consistency.
4. In chapter 5, we present a novel two-layer MRFs framework in which stereo matching and surface boundary estimation are combined. Both properties are inferred simultaneously and globally so that they can benefit each other. This work has direct application in phosphene vision based human indoor navigation.
5. In chapter 6, we give a summary of our work and list main contributions. We end the chapter by discussing some promising directions for future research.

Chapter 2

Theoretical Backgrounds and Related Work

2.1 Bayesian Labeling and Markov Random Field

2.1.1 Labeling Problems

A variety of problems in computer vision can be formulated as labeling in which the optimal solution is defined as the maximum probability estimation. And these problems are commonly referred to as *labeling problems*. They widely exist in early vision tasks like image segmentation, stereo matching, image restoration, texture synthesis and others, as shown in Figure 2.1. Labels represent different meanings in different tasks, for instance, intensity values as in grayscale image restoration and depth values as in stereo matching.

More formally, let \mathbf{L} be a set of n discrete labels.

$$\mathbf{L} = \{1, 2, \dots, n\} \quad (2.1)$$

And assume we have a set of discrete variables \mathbf{X} defined over a lattice,

$$\mathbf{X} = \{1, 2, \dots, m\} \quad (2.2)$$

And labeling is to assign labels from \mathbf{L} to each random variable $X_i \in \mathbf{X}$, different variables can take different labels. Any possible assignment is called a *labeling configuration* (denoted by f). It can be clearly observed that the set F of all configurations takes values from

$$\mathbf{F} = \underbrace{n \times n \cdots \times n}_{m \text{ times}} = n^m \quad (2.3)$$

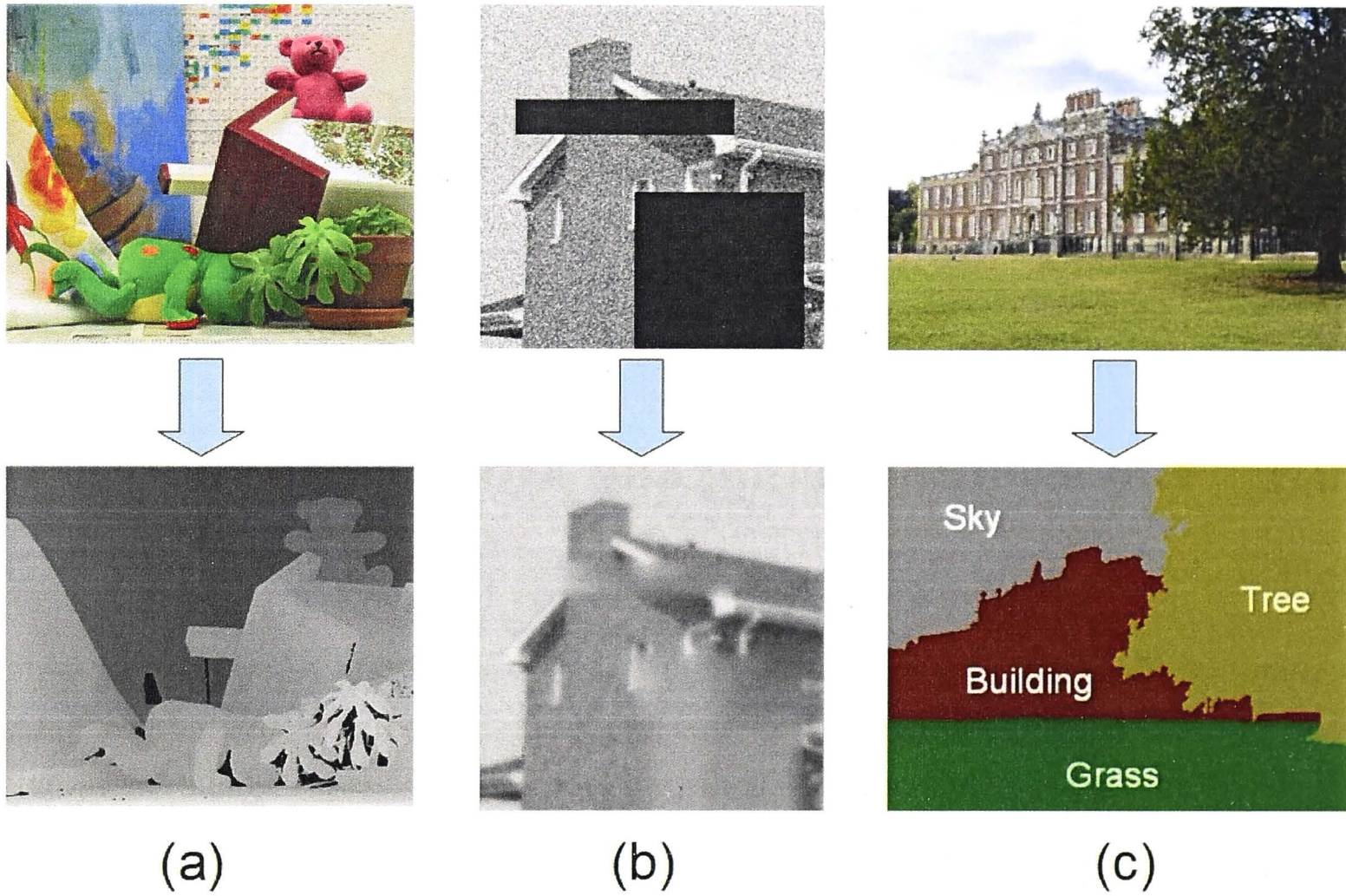


Figure 2.1: Some labeling problems in computer vision. (a) Stereo Matching: Given a pair of rectified left and right images, the depth information can be computed by finding the pixel correspondence along epipolar lines. The labels in this application represent depth values. (b) Image Denoising: Given a noisy image of the scene, the object is to infer the true intensity of the original image, here the labels are the intensities. (c) Object Class Segmentation: Given a observed image, the task is to divide the scenes into segmentations with semantic meanings. Here the set contains object labels $\{\text{sky, tree, building, grass}\}$

In terms of the maximum likelihood of the estimation of the true labeling, if we define r as the observation of the image, and we have *a posterior* probability $P(f|r)$ over a certain configuration f , then the goal is equivalent to maximize this probability and is called maximize *a posterior*(MAP) probability. The target to find the most probable labeling is to find the MAP configuration f^* that

$$f^* = \operatorname{argmax}_{f \in F} P(f|r) \quad (2.4)$$

Suppose we know both the *a priori* probability $P(f)$ and the probability densities $p(r|f)$ of the observation r , since the density function $p(r)$ does not affect the MAP solution, this posterior probability can be converted to a simple calculation using the Bayesian rule

$$P(f|r) = p(r|f)P(f)/p(r) \quad (2.5)$$

The likelihood function $p(r|f)$ is case wise, depending on the specific problems and will be discussed later, as knowing $P(f)$ is generally difficult. And this is the reason why Markov Random Field is introduced.

2.1.2 Markov Random Field

Markov Random Field(MRF) is a widely used probabilistic models described by an undirected graph for analyzing spatial or contextual dependencies of physical phenomena.[44]. Here we will briefly review the Markov property shared by variables in a MRF.

Pairwise Markov Property: Any two non-adjacent variables are conditionally independent given all other variables: $x_i, x_j | X_{\setminus \{i,j\}}$

Local Markov Property: A variable is conditionally independent of all other variables given its neighbors: $x_i, X_{\setminus \{i\}} | X_{neighbor\{i\}}$

Global Markov Property: Any two subsets of variables are conditionally independent given a separating subset: $X_A, X_B | X_S$ where every path from members in A to members in B passes through S .

In other words, a probabilistic model is considered a MRF with respect to the joint probability distribution over a set of random variables if and only if

separation in the model implies conditional independence. Therefore, if define a neighborhood system as $N = \{N_x | x \in X\}$, then a Markov Random Field satisfies

$$p(x_i | X_{\setminus \{i\}}) = p(x_i | x_j : j \in N_i) \quad (2.6)$$

According to the Hammersley-Clifford theorem[40], the posterior distribution $P(r|f)$ over the labelings of a MRF is a *Gibbs* distribution and can be written as

$$P(r|f) = \frac{1}{Z} \exp\left(-\sum_{c \in C} \Psi_c(\mathbf{X}_c)\right) \quad (2.7)$$

where Z is a normalizing constant known as the partition function, and C is the set of all cliques in the MRF, and $\Psi_c(\mathbf{X}_c)$ are potential functions defined over cliques c . In definition, a clique is a set of nodes that in which any pair of two nodes are adjacent in the MRF. The corresponding Gibbs energy is given by

$$E(x) = -\log P(r|f) - \log Z = \sum_{c \in C} \Psi_c(\mathbf{X}_c) \quad (2.8)$$

Since Z is a constant with respect to different labeling configurations, maximum *a posteriori*(MAP) labeling f^* is equivalent to the minimum of the Gibbs energy.

$$f^* = \operatorname{argmax}_{f \in F} P(r|f) = \operatorname{argmin}_{f \in F} E(x) \quad (2.9)$$

For more details, please refer to [8].

Orders and Structures of Markov Random Field

Based on the largest clique size c in the Equation 2.8, the MRF are sorted into two categories, second-order and higher-order. The second-order MRF are commonly referred to as pairwise MRF in which the largest clique size is 2. And for MRFs with clique size larger than 2, they are known as higher-order MRFs.

Pairwise model has been widely used in computer vision[69] due to its good enforcement of spatial coherence and efficiency of implementation. If we denote $x_i \in X$ as the hidden variables, $y_i \in Y$ as the corresponding observed variables, and $x_i, x_j \in N$ are two neighboring hidden variables, then the joint distribution of a pairwise MRF can be written as:

$$p(x, y) = \frac{1}{Z} \prod \phi_i(x_i, y_i) \prod \psi_{i,j \in N}(x_i, x_j) \quad (2.10)$$

Here, $\phi_i(x_i, y_i)$ comes from the likelihood of local measurement, and $\psi_{i,j \in N}(x_i, x_j)$ is usually defined as a prior enforcing consistency of adjacent variables.

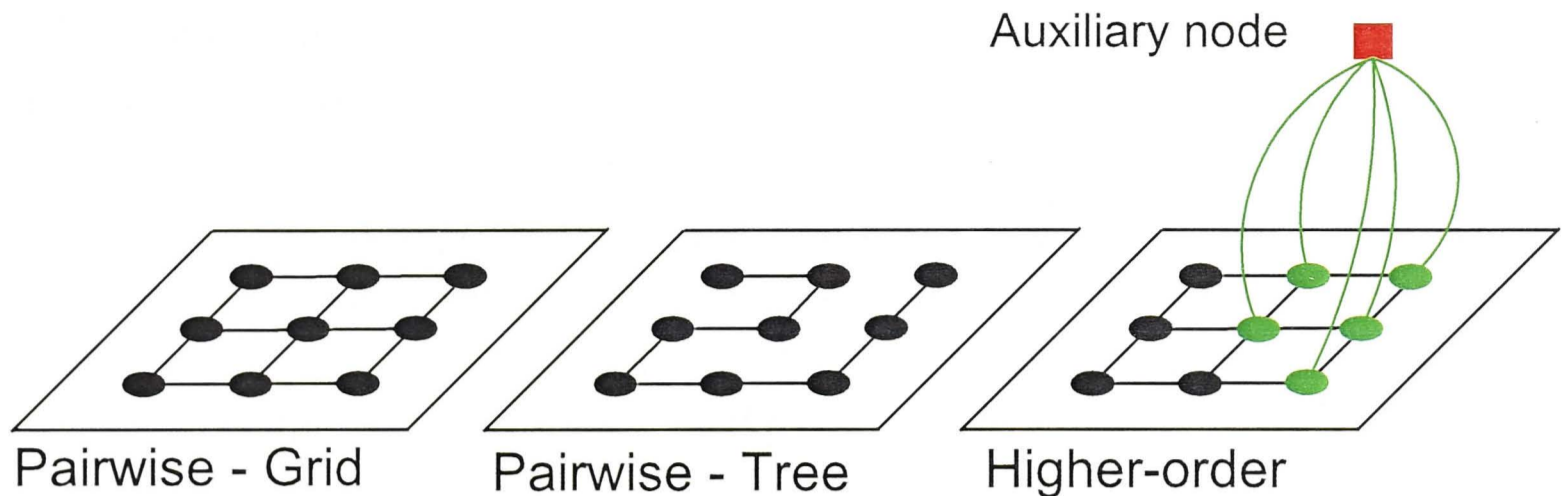


Figure 2.2: From left to right, the first one is the grid pairwise structure and it is the most common structure in computer vision; the second one is also a pairwise MRF, but it is loop-free; the third one is a higher-order MRF with maximum size of 5. In higher-order graph, the extra red node is a auxiliary node that connects every member of the clique. This expression is equivalent to fully connection of all members.

Although pairwise MRFs are generally easy to optimize, they are incapable of encoding the relationship between a group of variables. To overcome it, researchers have developed higher-order MRFs. For example in [35], segment is modeled as one clique in which its members are fully connected. And in [59], each boundary piece is taken as one node, and the conjunction of boundaries is modeled as the higher-order connection of these boundary nodes.

For fully connected groups of pixels, the joint distribution of its probability can be written as:

$$p(x, y) = \frac{1}{Z} \prod \phi_i(x_i, y_i) \prod \psi_{i,j,k \dots \in C}(x_i, x_j, x_k \dots) \quad (2.11)$$

where $\prod \phi_i(x_i, y_i)$ remains the local measurement, and $\prod \psi_{i,j,k \dots \in C}(x_i, x_j, x_k \dots)$ encodes the potential defined on a clique C .

An illustration of these classic structures is given in Figure 2.2.

2.2 Inference

After defining the Markov Random Field(MRF), the natural question raises, how can we infer the labeling that maximize *a posterior* estimation or minimize the energy function of a MRF?

The energy minimization algorithms originally used in 1990's were computational inefficient or ineffective, such as iterated conditional modes(ICM)[6] or

simulated annealing[4]. Over the past decade, energy minimization approaches have had a renaissance, novel algorithms have been developed such as graph cut[11][38] and loopy belief propagation[79]. It results in the prosperity of a variety of approaches using energy minimization to solve computer vision problems.

Here we will briefly review some of the representative energy minimization algorithms.

2.2.1 Iterated Conditional Models(ICM)

The iterated conditional models(ICM) known as one of the classic “greed” strategy based algorithms is firstly introduced in [6]. It starts by an initial labeling, and optimize each variable by choosing the label that decrease the most amount of energy. The advantage of this algorithm is that it is guaranteed to converge. The shortages are obvious too, it is extremely sensitive to the initial labeling especially in high-dimensional spaces with nonconvex energies and easily to stuck in local minimums. Therefore ICM has not been widely used in computer vision.

2.2.2 Graph Cut

Graph cut has been intensively explored during the past decade. It is firstly introduced into computer vision early in [28]. It is a algorithm of computing minimum for binary labeling. It first converts a MRF to a graph, every potential defined on the MRF becomes weight on the graph, and then it optimize the graph by finding a minimum cut using max-flow algorithm. It is guaranteed to achieve the global minimum when certain requirements are met.

However most of the labeling tasks in computer vision have multi labels. Therefore in [11], two algorithms $\alpha\beta$ – *swap* and α – *expansion* are proposed, the usage of graph cut is extended from binary to multi-label. For both algorithms, it lowers its energy by using binary graph cut as an inner loop, and it converges when no lower energy can be found.

For $\alpha\beta$ – *swap*, in each inner iteration, two random labels from the labels set are taken as the current α and β . The binary cut only applies on the variables with the current label of either α or β . The variables with current label α can be swapped to β in this process, vice versa. The swap moves find the local minimum such that there is no swap move for any pair of labels α, β that will lead to a lower energy.

The α – *expansion* is applied in an analogously way. In each inner iteration,

one random label is taken as α . For the variables with current labels other than α , they will be involved in the binary cut in which their labels can be changed to α .

The advantages of graph cut are its effective and fast convergence. But certain requirements have to be satisfied in order to use it. Define a label set L and $\{\alpha, \beta, \gamma\} \in L$. For each pair of neighboring pixels $\{x_i, x_j\}$, it has a second-order energy potential ψ_{ij} . ψ_{ij} is called a *metric* if it satisfies

$$\psi_{ij}(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta \quad (2.12)$$

$$\psi_{ij}(\alpha, \beta) = \psi_{ij}(\beta, \alpha) \geq 0 \quad (2.13)$$

$$\psi_{ij}(\alpha, \beta) \leq \psi_{ij}(\alpha, \gamma) + \psi_{ij}(\gamma, \beta) \quad (2.14)$$

for any labels $\{\alpha, \beta, \gamma\} \in L$. If ψ_{ij} only satisfies 2.12 and 2.13, it is called a *semi-metric*.

Originally in [11], α -*expansion* can only be applied when it is a metric, and semi-metric for $\alpha\beta$ -*swap*. In the later work [38], it relaxes these constraints and show that the expansion-move can be used when

$$\psi_{ij}(\alpha, \alpha) + \psi_{ij}(\beta, \gamma) \leq \psi_{ij}(\alpha, \gamma) + \psi_{ij}(\beta, \alpha), \quad (2.15)$$

and the swap-move algorithm can be used if

$$\psi_{ij}(\alpha, \alpha) + \psi_{ij}(\beta, \beta) \leq \psi_{ij}(\alpha, \beta) + \psi_{ij}(\beta, \alpha). \quad (2.16)$$

And we refer to these constraints as submodular conditions. When these conditions are not satisfied, graph cut algorithm can still be applied by truncating the violating terms[64], the deterioration degree will depend on the number of terms need to be truncated.

More details about graph cut will be discussed in the next section.

2.2.3 Loopy Belief Propagation(LBP)

Belief propagation(BP) is a powerful inference engine. The principles of it are clearly explained in [80]. It is based on iterative message passing. In every iteration, every node updates its message based on its local evidence and received message from the last iteration, and further passes this updated message to its neighbors according to the pre-defined graph structure. According to different usage, BP can be sorted into two categories, max-product based and sum-product based. Sum-product BP computes the marginal probability distribution of each

nodes in the graph. The commonly used one is the max-product BP, because for most of the tasks in computer vision, the optimization goal is to find the labeling with the MAP or the lowest energy, and this is exactly what max-product BP aims at.

In the original design[53], BP is for graphs without cycles. However this is not the case in computer vision, even the simplest pairwise MRF is with loops. Therefore researchers have developed a variant of BP, loopy belief propagation(LBP), and successfully applied on loopy graphs. Later in [20], researchers have greatly improved its efficiency by three modifications of distance transform, chessboard updating and hierarchical network.

Let $M_{i \rightarrow j}^t$ be a message that variables i sends to its neighbor j at iteration t . Then message updating rule of a typical pairwise MRF is

$$M_{i \rightarrow j}^t(x_i) = \min(\phi(x_i) + \sum_{k \in N(i)/j} M_{k \rightarrow i}^{t-1}(x_i) + \psi_{ij}(x_i, x_j)) \quad (2.17)$$

Generally, LBP is not guaranteed to converge and may stuck in an infinite loop, but for most tasks in early vision, it gives adequately good results and is widely applied.

2.2.4 Dynamic Programming(DP)

Dynamic programming(DP) is a algorithm for solving complex problems by breaking it down into a sequence of simpler subproblems. In computer vision, it is firstly used in finding the corresponding points along each epipolar line in stereo matching[50].

However, when in a graph without loops, DP is equivalent to belief propagation. Researchers have taken advantage of this and performed DP in well modeled tree structure(tree structures are naturally acyclic).[71]

2.2.5 Tree-Reweighted Message Passing(TRW)

Tree-reweighted message passing(TRW) is originally proposed in[72]. The key idea is tree-based relaxation, using a convex combination of tree structured distribution to derive the lower bounds on the energy of the MAP configuration.

Similar to LBP, the message that variable i sends to its neighbor j at iteration t is defined as

$$M_{i \rightarrow j}^t(x_i) = \min(c_{ij}(\phi(x_i) + \sum_{k \in N(i)/j} M_{k \rightarrow i}^{t-1}(x_i)) + \psi_{ij}(x_i, x_j) - M_{j \rightarrow i}^{t-1}(x_i)). \quad (2.18)$$

A set of trees are defined over the graph connections so that each edge will be included in at least one tree. And the coefficient c_{ij} is determined by the probability of the edge $\{x_i, x_j\}$ contained by a randomly chosen tree. If c_{ij} is set to 1, then it is identical to LBP, therefore it is a generalization of LBP.

However the original TRW algorithm does not guarantee to converge and the increment of lower bounds with iterations does not necessary occur. Later a variant called TRW-S[37] is proposed to overcome this shortage, in which the lower bound is promised not to decrease, resulting in a convergence property. It is the most often used version in practice.

2.3 Pseudo-boolean Optimization and Graph Cuts

2.3.1 Graph Cuts in Computer Vision

Graph cuts remains one of the active research areas in the past decade. Many of the tasks in computer vision can be formulated as an energy minimization problem, and graph cuts has been used as one of the major optimization tools under this purpose. It has been used in a wide variety of low-level vision applications, such as image denoising, image segmentation, image synthesis, stereo matching and so on. Beyond finding new applications, the researchers also obtained huge progress in itself, including efficient max-flow algorithms, constraint, multi-label problem and so on.

Although graph cuts has been firstly introduced into computer vision early in 1989 by Greig[28], but the real milestone are two classic papers written by Boykov[11][10] in 1999 and 2001 respectively. Paper[11] successfully introduces two algorithms that expand the ability of graph cuts from binary to multi-label, namely expansion and swap, and is the beginning of broad usage in computer vision. Paper[10] not only compares two common max-flow algorithms, but also introduced an improved version of the augmented-path. Later Kolmogorov states the well-known “sub-modularity” problem[38] as the essential constraint in graph cuts, and further expands second-order to third-order. In [36], Kohli proposed the idea that search trees can be re-used in order to achieve higher performance. The popular graph cuts tool we use nowadays in vision is the combination of these papers.

Another interesting work[24] appears later and manages to link graph cuts to pseudo-boolean optimization, and further extends third-order to higher-order. Based on it, researches have given theoretical prove on general transformation of

higher-order terms to second-order terms by adding auxiliary nodes[30], but in such conversion exponential auxiliary nodes in the worst case are needed which make it unsuitable in real use. To overcome it, researchers have explored and proposed some specific form of higher-order energy potentials, such as the classic P^n Potts Model in [34] and Robust P^n Potts Model in [35]. After that, a sparse and efficient generalization of Robust P^n Potts Model is given in [62], it can deal with lower-envelope higher-order terms. It is further extended to constrained upper-envelope higher order functions[33][27].

Meanwhile, some researchers focus on other aspects of graph cuts as well. Firstly, how to minimize un-submodular terms using graph cuts. In [63] author introduces techniques from pseudo-boolean optimization and names it QPBO, it can be used in non-submodular problems, after such optimization some of the labels may remain unlabeled. Secondly, efficiency. Researchers propose the Fusion Cuts[41], it decomposes the label space by 2-bits coding and minimizes them iteratively. Due to its parallel computation capability, it is adopted in many applications with large label space[9][48]. Thirdly, exact inference. There are some classic works on the exact inference including [58], but the restriction and calculation efficiency issue limit its usage. Fourthly, other models. Paper[15] introduces the new hierarchical model which is the extension of Kohli's Robust P^n Potts. Recently a joint two-layer MRF model is presented in[39]. Fifthly, similar pseudo-boolean optimization. In[13], author establishes the new framework that directly takes advantage of pseudo-boolean optimization, they start by eliminate the central pixel and build the new connection between its four neighboring pixels, and use approximation to simplify the higher-order term. The advantage is that it does not have to be regular, the drawbacks are its efficiency and lack of guarantee on approximation.

2.3.2 Pseudo-boolean Representation

To better understand the mechanism of graph cuts, we will firstly brief introduce pseudo-boolean representation here. Define variables $X = \{x_1, x_2, \dots, x_m\}$ taking values from $B = \{0, 1\}$, a pseudo-boolean function is a mapping

$$f : B^n \rightarrow R. \quad (2.19)$$

There are three ways to represent a pseudo-boolean function, namely Tableau, Posiform and Polynomial.

1. Tableau

In a Tableau form, it lists all 2^n values. For example, $f(x) =$

x_1	x_2	x_3	term	value
0	0	0	$\bar{x}_1\bar{x}_2\bar{x}_3$	-1
0	0	1	$\bar{x}_1\bar{x}_2x_3$	-1
0	1	0	$\bar{x}_1x_2\bar{x}_3$	3
0	1	1	$\bar{x}_1x_2x_3$	2
1	0	0	$x_1\bar{x}_2\bar{x}_3$	-1
1	0	1	$x_1\bar{x}_2x_3$	-2
1	1	0	$x_1x_2\bar{x}_3$	5
1	1	1	$x_1x_2x_3$	1

And it is equivalent to

$$f(x) = -\bar{x}_1\bar{x}_2\bar{x}_3 - \bar{x}_1\bar{x}_2x_3 + 3\bar{x}_1x_2\bar{x}_3 + 2\bar{x}_1x_2x_3 - x_1\bar{x}_2\bar{x}_3 - 2x_1\bar{x}_2x_3 + 5x_1x_2\bar{x}_3 + x_1x_2x_3 \quad (2.20)$$

2. Posiform

If we replace the terms with negative coefficient in the above equation, for example

$$\begin{aligned} -\bar{x}_1\bar{x}_2\bar{x}_3 &= -(1-x_1)\bar{x}_2\bar{x}_3 \\ &= x_1\bar{x}_2\bar{x}_3 - (1-x_2)\bar{x}_3 \\ &= x_1\bar{x}_2\bar{x}_3 + x_2\bar{x}_3 - (1-x_3) \\ &= x_1\bar{x}_2\bar{x}_3 + x_2\bar{x}_3 + x_3 - 1, \end{aligned} \quad (2.21)$$

so that every all coefficients are positive, then it is a Posiform. More formally, let $u_i = \{x_i, \bar{x}_i\}$, then a third-order pseudo-boolean function can be expressed in Posiform as

$$f(x) = a_0 + \sum_i a_i u_i + \sum_{i,j} a_{ij} u_i u_j + \sum_{i,j,k} a_{ijk} u_i u_j u_k, \quad (2.22)$$

where all a_i , a_{ij} and a_{ijk} are positive.

Note that Posiform representation is not unique, for instance, $x_i\bar{x}_j = -1 + x_1 + \bar{x}_2 + \bar{x}_1x_2$.

3. Polynomial

Similarly, if we replace every \bar{x}_i by $1-x_i$, for example

$$\begin{aligned} -\bar{x}_1\bar{x}_2\bar{x}_3 &= 1(1-x_1)(1-x_2)(1-x_3) \\ &= x_1x_2x_3 - x_1x_2 - x_1x_3 - x_2x_3 + x_1 + x_2 + x_3. \end{aligned} \quad (2.23)$$

It is in Polynomial form. A typical Polynomial form of a third-order pseudo-boolean function can be denoted as

$$f(x) = c_0 + \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i x_j + \sum_{i,j,k} c_{ijk} x_i x_j x_k. \quad (2.24)$$

And Polynomial representation is unique.

2.3.3 Energy Cost and Graph Representation

Let variables $X = \{x_1, x_2, \dots, x_m\}$ take values in a binary label set B , and assume a neighborhood structure N_{x_i} so that

$$x_i \in N_{x_j} \Leftrightarrow x_j \in N_{x_i} \quad (2.25)$$

$$P(X_i = x_i | X_j = x_j; j \neq i) = P(X_i = x_i | X_j = x_j; j \in N_{x_i}) \quad (2.26)$$

where $P(X_i = x_i)$ represents the conditional probability distribution of a given variable x_i .

According to the Gibbs distribution in 2.8, finding the assignment that maximizes the probability is equivalent to minimize the corresponding energy function.

A classical energy function on a 4-connected MRF is defined as

$$E = \sum_{x_i} E(x_i) + \sum_{x_i x_j} E(x_i, x_j), \quad (2.27)$$

where the first and second terms are referred to as Data term and Smoothness term respectively. Date term is based on the local observation and applied on every node independently, while smoothness term usually acts as constraints to let neighboring nodes smoothed.

Let p, q be the boolean values 0 or 1, and $E_{i;p}$ to denote the cost when x_i takes the value p . Similarly, $E_{ij;pq}$ is incurred if $x_i = p$ and $x_j = q$. Therefore, the cost associated with two variables x_i and x_j are redefined in which

$$E(x_i) = E_{i;1}x_i + E_{i;0}\bar{x}_i \quad (2.28)$$

and

$$E(x_i, x_j) = E_{ij;00}\bar{x}_i\bar{x}_j + E_{ij;01}\bar{x}_ix_j + E_{ij;10}x_i\bar{x}_j + E_{ij;11}x_ix_j. \quad (2.29)$$

And the sum of the data term and smooth term cost can be formulated in an alternative form, namely quadratic function.

The standard form of a quadratic function is defined as,

$$f(x) = a_0 + \sum_i a_i u_i + \sum_{1 \leq i \leq j \leq n} a_{ij} \bar{x}_i x_j, \quad (2.30)$$

where $u_i = x_i$ or \bar{x}_i , and $a_i, a_{ij} \geq 0$.

In general, minimizing pseudo-boolean functions is a NP hard problem (with respect to the number of variables n). It can be observed that a pseudo-boolean function with n variables can have up to 2^n terms, thus leading to exponential time. A solution is to convert the pseudo-boolean functions to a graph, and use max-flow algorithms[10].

Here let $G = \langle V, E \rangle$ be a undirected graph with a set of vertices V and a set of directed edges E that connect them. V contains not only one-to-one correspondence from variables X but also two special *terminal* nodes, which are called the *source*(s) and the *sink*(t). Each edge is assigned some nonnegative weight $w(p, q)$, and $w(p, q)$ may differ to $w(q, p)$. An edge is called a $n - link$ if it connects two variables, the edge which connects a variable to a terminal will be called a $t - link$.

Every node (except for terminals of course) connects two both terminals at the beginning, a st-cut C is to partition the graph to two disjoint subsets S and T such that source s is in S and t is in T , and every variables will only remain one $t - link$. The cost of a cut C is the sum of pair of weight $w(p, q)$ the cut passing through if p and q do not remain the same $t - link$. The *minimum cut* problem is to find a cut that has the minimum cost among all cuts.

Here we specially examine the case of quadratic pseudo-boolean functions, and show how they could be converted to graph representation.

First, transform the quadratic function to the form of

$$f(x) = L + \sum_{ij} a_{ij} \bar{x}_i x_j, \quad (2.31)$$

where L represents linear terms of $x_i(\bar{x}_i)$ and constant a_0 .

Second, draw a graph with vertices which one-to-one corresponds to variables. Then assign edges as follows:

1. Draw *source*(s) and *sink*(t) to represent 0 and 1 respectively.
2. For the constant term a_0 , add an edge from *source* to *sink* with weight a_0 .
3. For a term $a_i x_i$, add an edge from *source* to x_i with weight a_i .
4. For a term $a_{\bar{i}} \bar{x}_i$, add an edge from x_i to *sink* with weight $a_{\bar{i}}$.
5. For a term $a_{ij} \bar{x}_i x_j$, add an edge from x_i to x_j with weight a_{ij} .

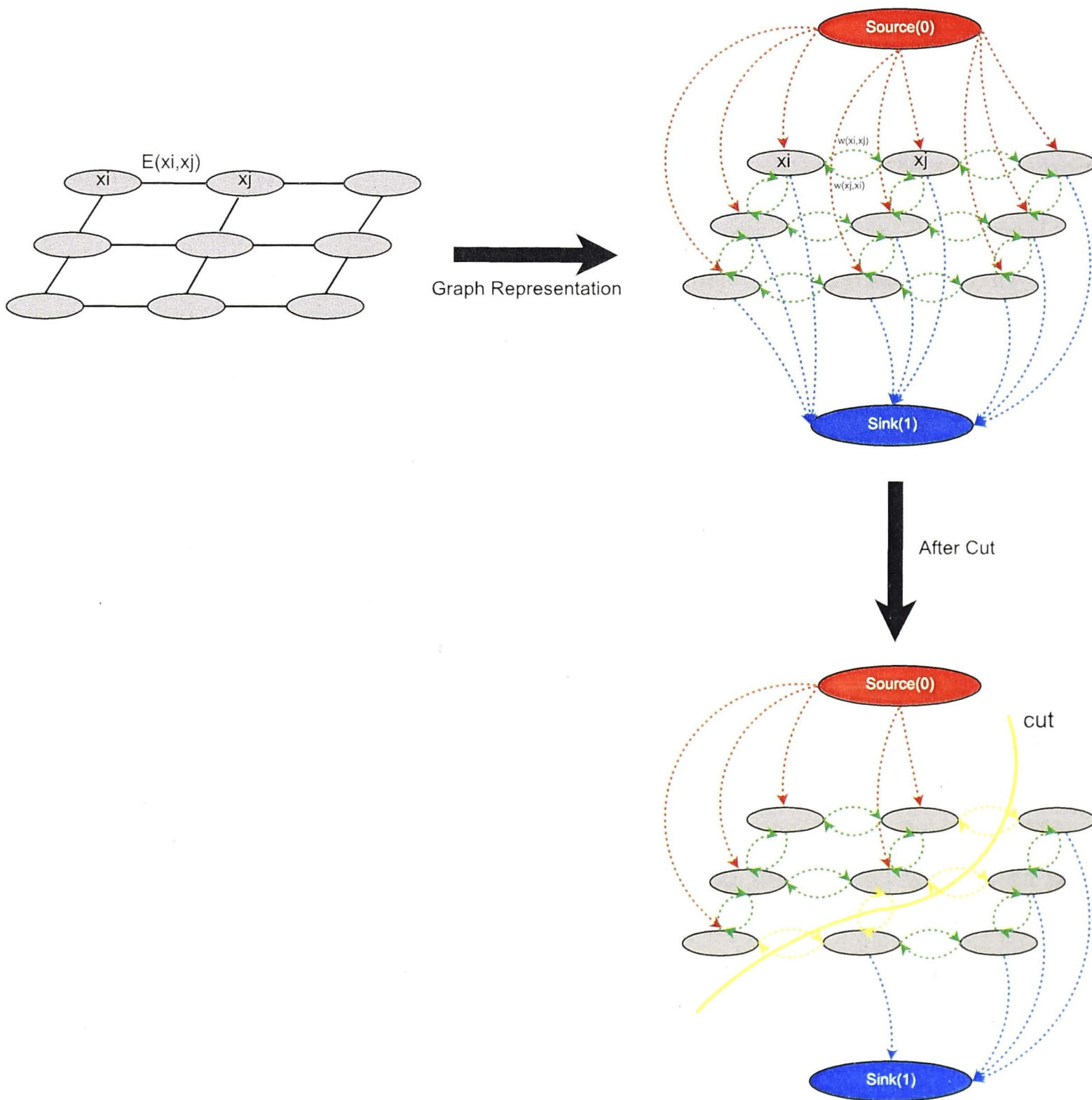


Figure 2.3: Graph construction on a MRF and its cut.

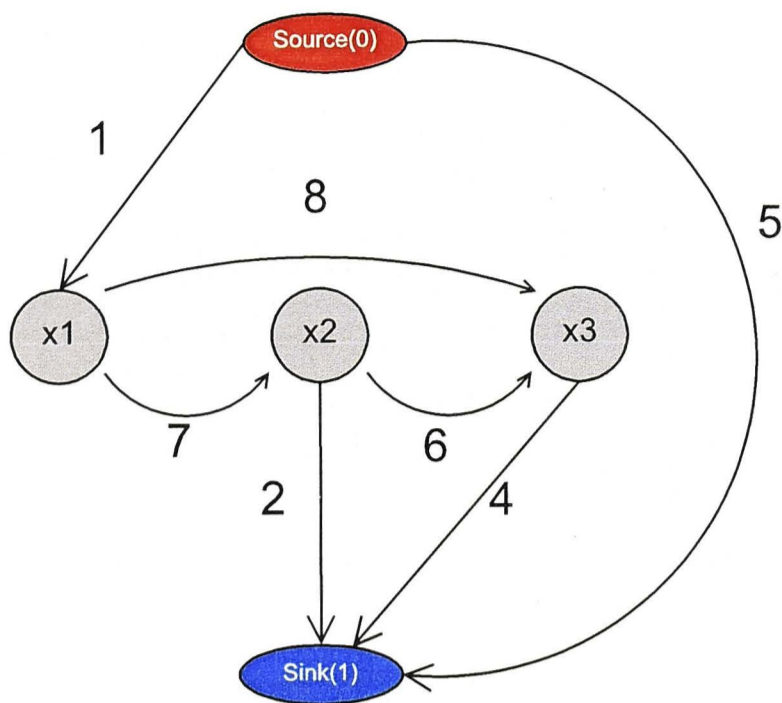


Figure 2.4: A graph representation example.

For example,

$$f(x) = 5 + x_1 + 2\bar{x}_2 + 4\bar{x}_3 + 7\bar{x}_1x_2 + 8\bar{x}_1x_3 + 6\bar{x}_2x_3 \quad (2.32)$$

can be converted to the graph shown in Figure 2.4.

A quadratic function $f(x)$ is called either *regular* or *submodular* depending on whether $a_{ij} \geq 0$ for all i, j in Equation 2.31. This constraint is called *submodularity constraint*. Only submodular quadratic functions are graph representable and can be further solved using max-flow algorithm.

Hence, the submodularity constraint of a second-order energy function can be conducted. According to Equation 2.27, the data term is $E(x_i)$ and the smoothness term is $E(x_i, x_j)$. The data term can always be the form as

$$E(x_i) = \begin{cases} a_i, & \text{if } x_i = 1, \\ a_{\bar{i}}, & \text{if } x_i = 0. \end{cases} \quad (2.33)$$

It can be further represented using the Posiform as

$$E(x_i) = a_i x_i + a_{\bar{i}} \bar{x}_i. \quad (2.34)$$

Similarly, the smoothness term can be converted to

$$E(x_i, x_j) = \begin{cases} a_{ij}, & \text{if } x_i = 1, x_j = 1, \\ a_{i\bar{j}}, & \text{if } x_i = 1, x_j = 0, \\ a_{\bar{i}j}, & \text{if } x_i = 0, x_j = 1, \\ a_{\bar{i}\bar{j}}, & \text{if } x_i = 0, x_j = 0. \end{cases} \quad (2.35)$$

And it can be further represented as

$$E(x_i, x_j) = a_{ij}x_ix_j + a_{i\bar{j}}x_ix_{\bar{j}} + a_{\bar{i}j}\bar{x}_ix_j + a_{\bar{i}\bar{j}}\bar{x}_i\bar{x}_j. \quad (2.36)$$

The data term is a first-order linear term and the coefficient can always be transformed to positive through simple variable substitution and hence is always graph representable.

On the other hand, smoothness may violate the submodularity. Its constraint can be derived in this way.

$$\begin{aligned} E(x_i, x_j) &= a_{ij}x_ix_j + a_{i\bar{j}}x_ix_{\bar{j}} + a_{\bar{i}j}\bar{x}_ix_j + a_{\bar{i}\bar{j}}\bar{x}_i\bar{x}_j \\ &= a_{ij}(1 - \bar{x}_i)x_j + a_{i\bar{j}}(1 - \bar{x}_i)(1 - x_j) + a_{\bar{i}j}\bar{x}_ix_j + a_{\bar{i}\bar{j}}\bar{x}_i(1 - x_j) \\ &= L + (a_{i\bar{j}} + a_{\bar{i}j} - a_{ij} - a_{\bar{i}\bar{j}})\bar{x}_ix_j \end{aligned} \quad (2.37)$$

L is first-order term plus constant term, and is always submodular. For the rest of the term, it is graph representable only if

$$a_{i\bar{j}} + a_{\bar{i}j} - a_{ij} - a_{\bar{i}\bar{j}} \geq 0, \quad (2.38)$$

and this conclusion is the classic submodular constraint for quadratic pseudo-boolean functions or second-order energy functions.

More specifically, in alpha-expansion, 0 represents the variable keep its current label, and 1 represents the variable taking the expandable label α . If we define the current labels of two neighboring variables as p and q (note alpha-expansion will only be applied when $p, q \neq \alpha$), and use $\psi(p, q)$ to denote the smoothness energy function, then the submodularity for alpha-expansion becomes

$$\psi(p, \alpha) + \psi(\alpha, q) - \psi(p, q) - \psi(\alpha, \alpha) \geq 0. \quad (2.39)$$

Similarly, for alpha-beta-swap, suppose the current swap pairs are p and q ($p \neq q$), two neighboring nodes with current labels as either p or q will participate in this process. If define 0 as the potential new label p , and 1 for q , then the constraint is

$$\psi(p, q) + \psi(q, p) - \psi(p, p) - \psi(q, q) \geq 0. \quad (2.40)$$

2.3.4 Max-Flow Algorithms

Once the graph representation step has been done, the next step is to compute the minimum cut. It has been proven in combinatorial optimization that finding

the *minimum cut* is equivalent to finding the *maximum flow* from the source s to the sink t , and in fact these two values are equivalent as well.

There exist many polynomial time algorithms for min-cut/max-flow[52][10]. Generally, these algorithms can be sorted into two main groups: “push-relabel” style methods[25] and “augmenting path” style methods[23]. For push-relabel methods, there is no valid flow during the operation, instead there are “active” nodes with a positive “flow excess”. While augmenting-paths based algorithms work by pushing flow along non-saturated paths from the source to the sink until the maximum flow in the graph is reached. Another advantage of push-relabel algorithms is parallel computable over graph nodes, therefore it can be accelerated by GPU which is a very promising direction for real-time application. However, in computer vision applications, the most common used algorithm currently is the one presented in [10] which is a fast version of augmenting-paths. We refer it to “new max-flow algorithms”.

Traditional augmenting-paths based techniques need a search tree for breadth-first search, however it is computational expensive, which makes it unusable in practice. Therefore, in the new max-flow algorithms, authors in [10] develop a new min-cut/max-flow algorithm based on augmenting paths. In terms of building search trees for detecting augmenting paths, they build two search trees, one from the source and the other from the sink which greatly speed up the process. Moreover, two search trees can be reused and do not need to be rebuilt every time, however the drawback is that the found path is not necessarily the shortest path. Theoretically speaking, The computational complexity of the new algorithm is worse than the standard algorithms, but the authors prove that it significantly outperforms standard algorithms.

The New Max-Flow Algorithm Overview

Here we will briefly introduce the new max-flow algorithms, because it is the key to graph cuts optimization in my vision applications. There are two trees S and T with roots at s and t respectively. There are two types of nodes, the one that locates on the out border and can further grow by acquiring new children are called “Active”(A), and the one that can not grow are named “Passive”(P). The algorithm iteratively repeats three main stages : growth stage, augmentation stage and adoption stage.

In the growth stage, the active nodes explore adjacent non-saturated edges and acquire new children from set of free nodes. Once all neighbors of a given active node are explored, the active node becomes passive. When active node encounters a neighboring node that belongs to the opposite tree, this stage terminates.

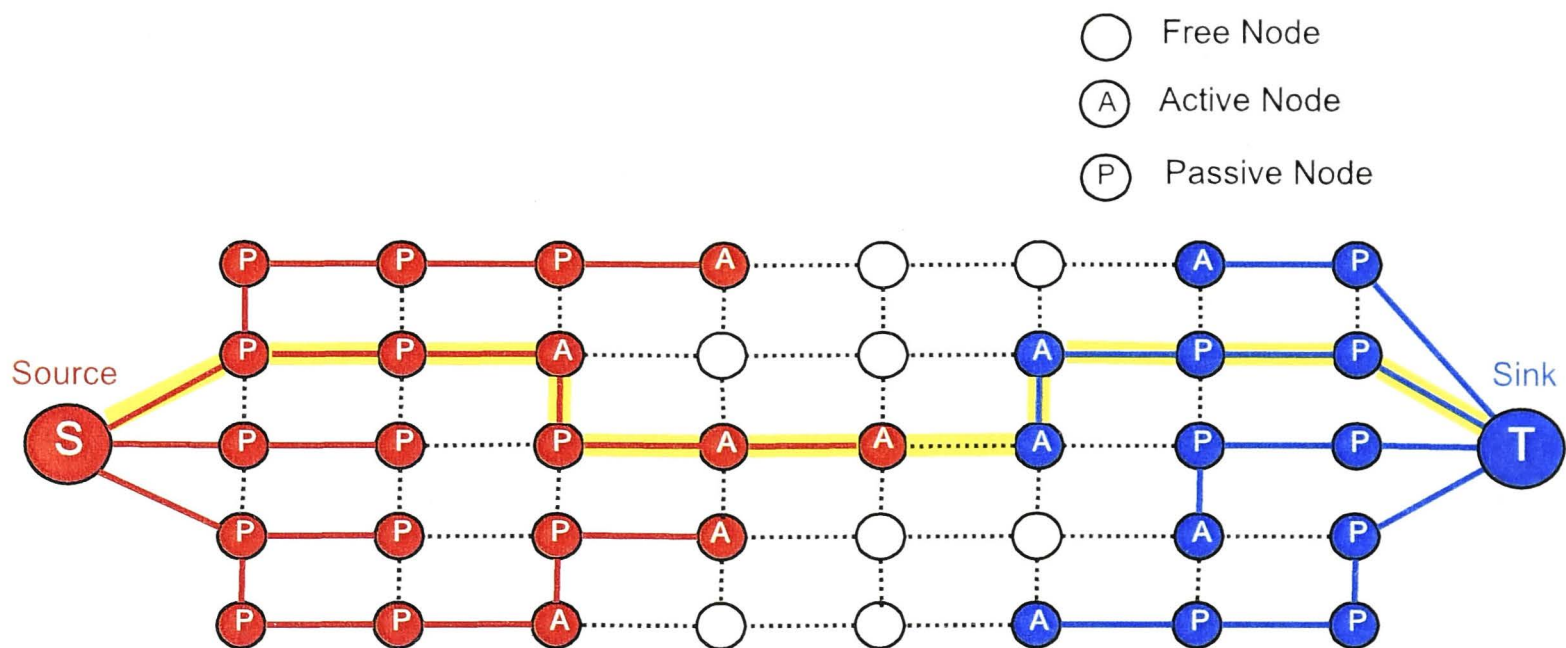


Figure 2.5: An example of the new max-flow algorithm.

In the augmentation stage, it saturates some edge(s) in the path by pushing through the largest possible flow. If the edge linking the children to their parents are saturated, then the edges are no longer valid, and the children become “orphans”. The result is, the augmentation phase may split the search trees into forests.

In the adoption stage, the algorithm restores tree structure by trying to find a new valid parent for each orphan. The requirement for the new parent is that it belongs to the same set (S or T) with the orphan and also connects the orphan with a non-saturated edge. If there is no qualified parent, then the orphan is removed from S or T and becomes a free node. It also denotes all its former children as orphans. When there is no orphans left, the adoption stage terminates. Thus the search trees of S and T are restored, as some orphan nodes in S and T may become free after this stage.

The algorithm iteratively do the three stages until the search tree S and T can not further grow (no active nodes) and the trees are separated by saturated edges which means a maximum flow is achieved.

After maximum flow is obtained, new labels of variables can be easily decided by examining which $t-link$ is left for each variable. For example in alpha-expansion, after the cut, nodes remain the $t-link$ to source will keep their current label, and those connect to sink, will take label α as their new label.

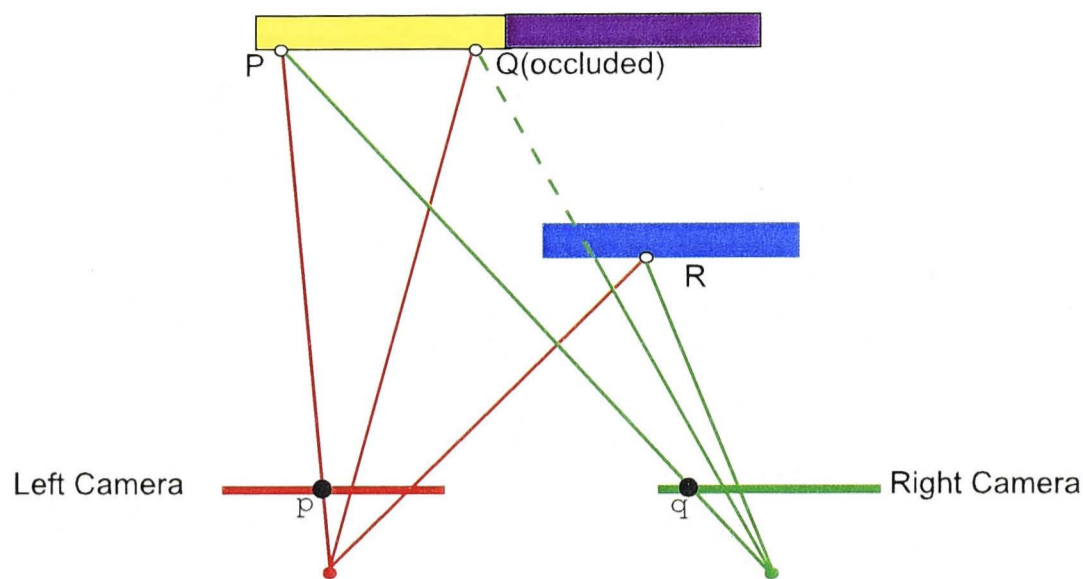


Figure 2.6: An example of two aligned cameras. The point P and R on the objects can be observed by two cameras at the same time, while the point Q only appears in the left camera and is occluded by the blue object from the right view. For point P , p and q are its projections in two cameras respectively.

2.4 Stereo Matching

2.4.1 The Two-Frame Stereo Matching Problem

Two-frame stereo matching has always been one of the most heavily researched topics in early vision problems. A few excellent comprehensive reviews can be found in [67][5][12][18]. Unlike human easily using their brain to perceive the depth, this task could be very challenging for computers.

The problem is often formulated as follow. A scene is captured by two cameras at the same time with known relative coordinate systems, the task is to determine a correspondence between each pixel p in the first image (also called the reference image, usually the left view) and some pixel q in the second image (usually the right view). That means, ideally a real point P of the scene has one projection pixel in each camera. The distance from the camera to the point P can be determined through simple computation. The reason we say “ideally” is that, there may exist the situation that some point P only appears in one camera but is occluded by some close objects in the other view. In this case, the distance can not be decided since there is no correspondence, and this situation is known as “occlusion” (Figure 2.6).

The most common two-camera setup in practice are that two aligned cameras differ only by a shift in the horizontal direction. To reduce the computation time, image rectification is applied on both images, therefore two corresponding pixels in left and right images are always on the same horizontal epipolar line. An

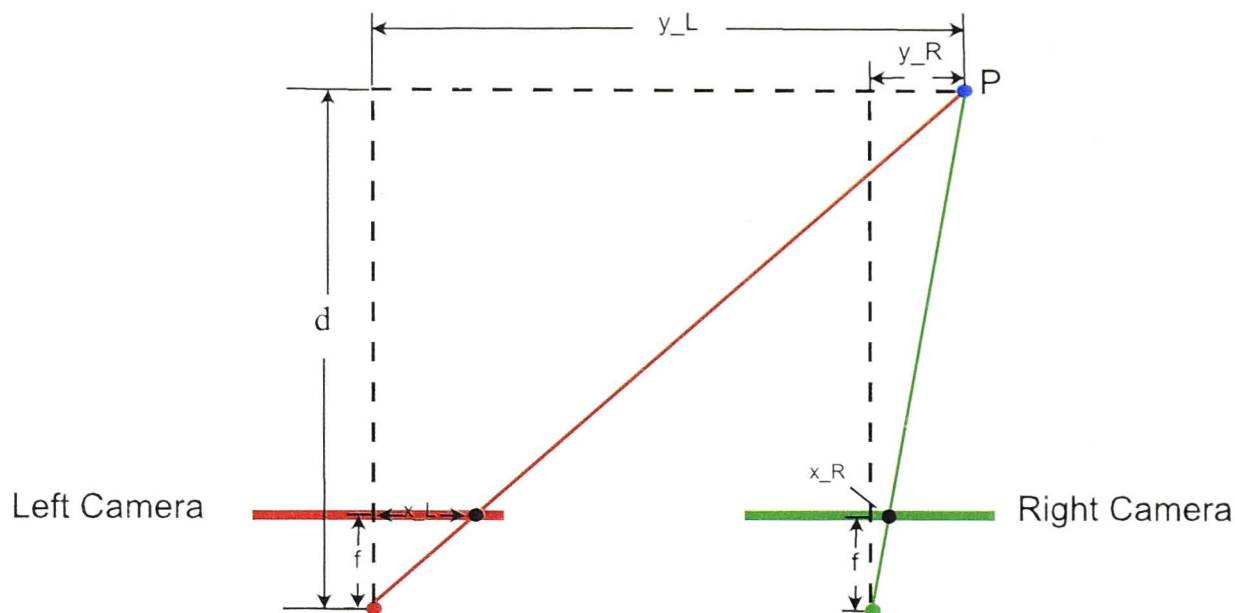


Figure 2.7: An example of how the horizontal distance between two corresponding pixels reflect the real distance.

illustration is given in Figure 2.7.

In Figure 2.7, it can be observed that

$$\begin{cases} \frac{x_L}{f} = \frac{y_L}{d} \\ \frac{x_R}{f} = \frac{y_R}{d} \end{cases} \quad (2.41)$$

After simple calculation, we can get

$$d = \frac{(y_L - y_R)f}{x_L - x_R} = \frac{lf}{x_L - x_R} \propto \frac{1}{x_L - x_R}. \quad (2.42)$$

In other words, the horizontal distance between two corresponding points ($x_L - x_R$) is inversely proportional to the actual distance from the cameras to the point in real world (d).

On the other hand, given the two perspective projection matrices $C = [q_{ij}]$ and $C' = [q'_{ij}]$, then for any scene point P with unknown 3D coordinates (X, Y, Z) , that projects onto the two camera at (u, v) and (u', v') , we have

$$C \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} \quad \text{and} \quad C' \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} s'u' \\ s'v' \\ s' \end{bmatrix}. \quad (2.43)$$

Eliminating s and s' and combining the two equations into matrix form gives

$$\begin{bmatrix} q_{11} - uq_{31} & q_{12} - uq_{32} & q_{13} - uq_{33} \\ q_{21} - vq_{31} & q_{22} - vq_{32} & q_{23} - vq_{33} \\ q'_{11} - u'q'_{31} & q'_{12} - u'q'_{32} & q'_{13} - u'q'_{33} \\ q'_{21} - v'q'_{31} & q'_{22} - v'q'_{32} & q'_{23} - v'q'_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} u - q_{14} \\ v - q_{24} \\ u' - q'_{14} \\ v' - q'_{24} \end{bmatrix}. \quad (2.44)$$

This is a linear system in (X, Y, Z) . The 3D coordinates of P can be easily computed.

It is worth noting that, researchers have generalized two-camera stereo to multi-camera, thus the ambiguity involved in matching can be further reduced. Also multi-camera have been applied successful in the application of scene reconstruction.

2.4.2 Matching Constraints

In order to minimize false matches, researchers usually impose some constraints in matching. Below is a list of the commonly used constraints.

Photoconsistency

For color(intensity)-based algorithms, if two pixels are corresponding to the same point in real world, then their colors(intensities) must be similar, this is sometimes referred to as Lambertian or constant brightness assumption. Similarly for feature-based approach, the matching features should share similar attribute values. The photoconsistency is the fundamental constraint in stereo matching, however it is sensitive to difference in camera gain or bias. Pair of cameras may have slightly different characteristics, and will result in different intensities. To overcome it, some algorithms use gradient-based or non-parametric measures instead.

Continuity

To against local ambiguities, spatially smoothness is commonly preferred. Unfortunately, this constraint does not hold for neighboring pixels across the depth surface boundaries, because depth could change abruptly there. Over-smoothing will lead to blur effect along surface boundaries.

Uniqueness

The uniqueness constraint has been applied as a hard constraint sometimes to minimize the risk of false matches. That is a given pixel from one image can match no more than one pixel from the other image. In other words, the uniqueness constraint enforces a one-to-one mapping between pixels in two images. However this constraint fails if there are transparent objects or occlusions.

Ordering

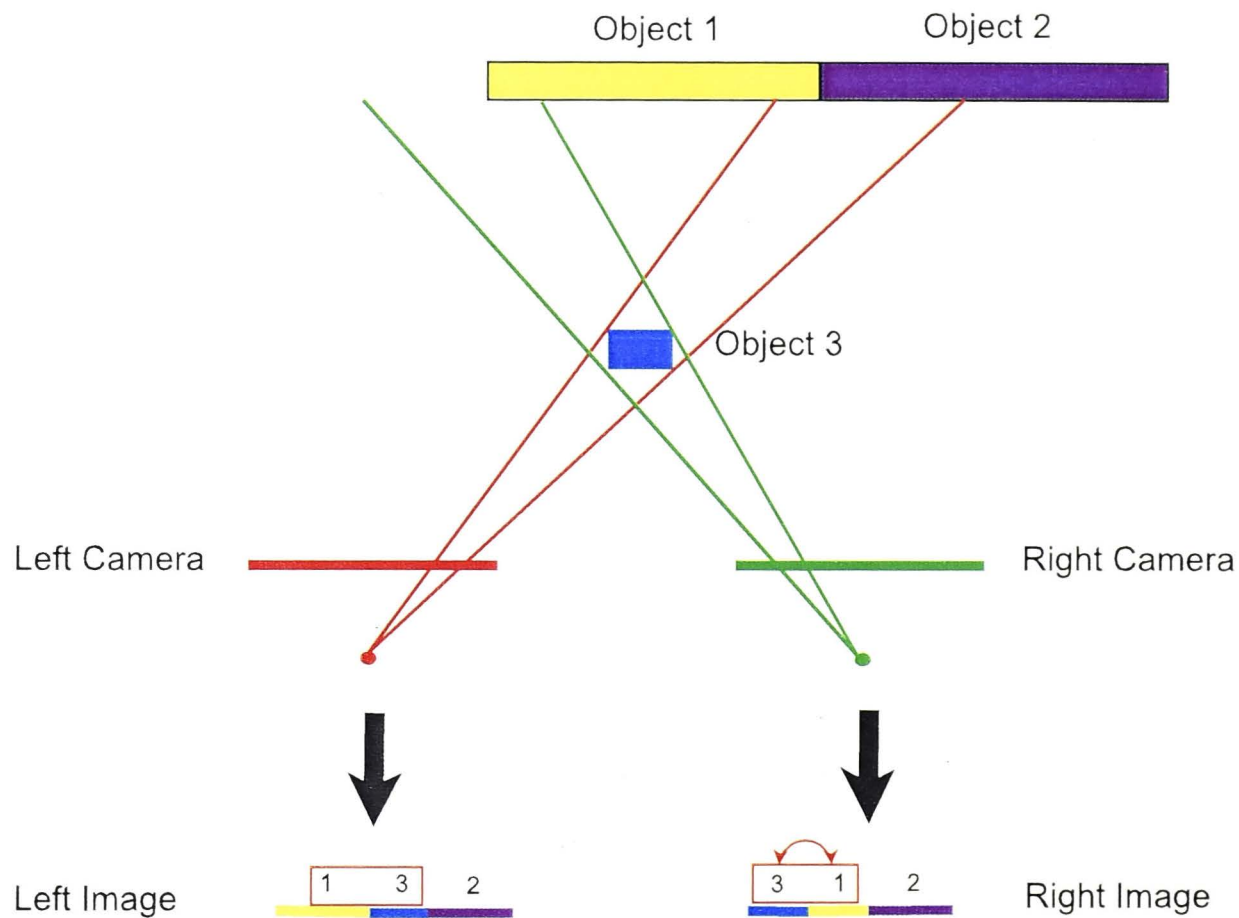


Figure 2.8: An example of violation of ordering constraint. In the left image, the object 1 is to the left of object 3, but the order is reversed in the right view.

The ordering constraint usually appears as a supplementary hard constraint to uniqueness. Two points P and Q , if P is to the left of Q in one view, then should remain the same order in the other view, vice versa. That is, the ordering of features is preserved order along scanlines in both input images. This constraint can be efficiently implemented by dynamic programming. It may be violated in practice though, a simple case is given in Figure 2.8.

2.4.3 Pixel-Based and Segment-Based Algorithms

Based on different representation of depth estimation, existing methods can be sorted into two categories: pixel-wise and segment-wise. Pixel-based algorithms are often suffering from local noises and being insufficient of the cues of the scene. As people generally identify the object and reconstruct the scene by partitioning the scene into a set of groups each with the same or similar visual features such as the color or texture, researchers have developed segment-based algorithms upon the similarity.

Segment-based algorithms have dominated the Middlebury Benchmark[67] due to their good performances on reducing ambiguity of disparities in texture-less regions. They usually share the assumption that the scene structure can be

approximated by a set of non-overlapping visually homogeneous regions where each region corresponds to its own depth surface. In other words, all pixels in the same segment should lie on the same depth surface and discontinuities only occur on boundaries. This assumption certainly enhances the tolerance of local noises as the depth surface is now decided by a group of pixels, the risk of assigning fault disparities to occluded or textureless individual pixels has been decreased. However, with segments being purely grouped on visually features, they are still likely to be influenced by local noises. Segment-based approach usually does not concern the dimension of the segment, and simplify each segment as an individual node in the model for further optimization. Therefore robustness will not be guaranteed due to the existing of those small segments.

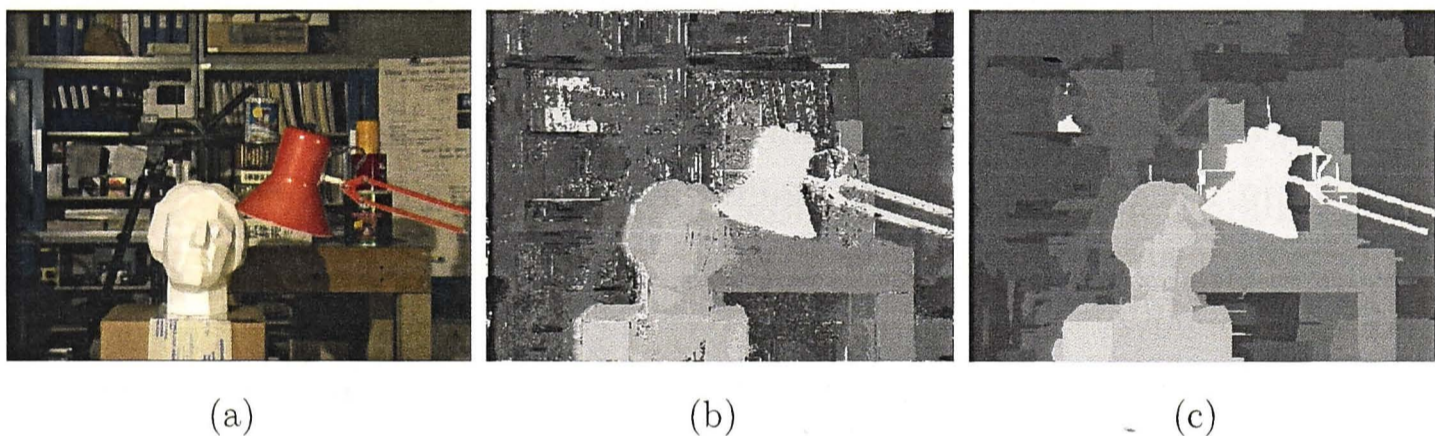


Figure 2.9: Typical results of pixel-based and segment-based algorithms, the data set is Tsukuba from Middlebury[67]. (a) original image, (b) pixel-based algorithm, (c) segment-based algorithm.

2.4.4 Local and Global Algorithms

Existing stereo algorithms can also be divided into local and global algorithms based on the optimization. Normally, both local and global algorithms have the same step of pixel-based matching cost.

The most common and easy matching cost algorithms are *sum of absolute difference*(SAD)[32], *sum of squared difference*(SSD)[1][29] and *normalized cross - correlation*(NCC)[65].

Define I_L and I_R as rectified left and right images, for every pixel p there is a two-dimensional support region w (usually a window) center at p . Then the cost for depth d at pixel p for sum of absolute difference is:

$$SAD : \quad cost(p, d) = \sum_{(m_x, m_y) \in w} |I_L(m_x, m_y) - I_R(m_x, m_y - d)|. \quad (2.45)$$

For sum of squared difference, the cost is:

$$SSD : \quad cost(p, d) = \sum_{(m_x, m_y) \in w} (I_L(m_x, m_y) - I_R(m_x, m_y - d))^2. \quad (2.46)$$

In the case of normalized cross-correlation, the cost is:

$$NCC : \quad cost(p, d) = \frac{\sum_{(m_x, m_y) \in w} I_L(m_x, m_y) * I_R(m_x, m_y - d)}{\sqrt{\sum_{(m_x, m_y) \in w} I_L^2(m_x, m_y) * \sum_{(m_x, m_y) \in w} I_R^2(m_x, m_y - d)}} \quad (2.47)$$

Besides these three, other traditional approaches include binary matching cost (i.e., match/ no match)[46] and the insensitive to difference in camera gain or bias ones, such as gradient-based measures[66] and non-parametric measures[82].

A widely used algorithm is described in [7] which is insensitive to image sampling. Instead of comparing pixel values by integral shift, this algorithm compare each pixel in the reference image against a linearly interpolated function of the other image. It achieves relatively good results while not sacrifice much on computation efficiency.

Usually the support region size in local matching cost is a size-fixed squared window. In practice, in order to achieve good results, window size should be set variously for different image pairs, and a perfect window size is always hard to tune. If the window size is too large, it will lose details, but too small will lead to more local noises. Therefore, researchers have developed algorithms with shiftable window[2] and adaptive window [51][31].

It is worth noting that researchers also improve the window-based aggregation by varying support-weights[81]. They adjust the support weights of the pixels in a given support window based on color similarity and geometric proximity to reduce the image ambiguity. It has one of the leading results among local algorithms in Middlebury benchmark.

More recently researchers found out by smoothing the matching cost volume with a efficient edge preserving filter, state-of-the-art results can be obtained[61]. In addition, this algorithms claims that it can be optimized to run in real-time.

Local Algorithms

Local approaches usually focus on matching cost computation and cost aggregation, once these steps have been done, the rest is trivial: usually a local “winner-take-all” (WTA) strategy is performed. That is, for every pixel p , choose d with the optimal $cost(p, d)$. Typically for SAD or SSD, the optimal cost is the one with the minimum value, and for NCC the maximum value would be the chosen

depth. Opposite to global algorithms, local algorithms neglect the smoothness of spatially neighboring pixels and the result is often not robust.

Global Algorithms

Unlike local algorithms, global approaches often formulate themselves as a pre-defined energy minimization problem in which the lowest energy corresponds to the optimal labeling. It iteratively minimizes the energy through some optimization techniques, once the energy cannot be decreased further or within a small threshold for certain times, then the process is terminated and returns the current labeling.

The standard form of a energy function is

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q), \quad (2.48)$$

where P is the set of pixels, and N is the neighboring system, and f denotes the labeling. $\sum_{p \in P} D_p(f_p)$ is referred to as *Data Term* (E_{data}), it usually measures how well the disparity f_p agrees with the input image pair. A typical choice for the data term is equal to the local matching cost, that is

$$E_{data} = \sum_{p \in P} D_p(f_p) = \sum_{p \in P} cost(p, f_p). \quad (2.49)$$

$\sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q)$ is called the *Smoothness Term* (E_{smooth}), and it often penalizes on difference between neighboring pixels ($\{p, q\} \in N$) to make the resulting depth smoothness.

A common form of E_{smooth} is

$$E_{smooth} = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) = \sum_{\{p,q\} \in N} \lambda |f_p - f_q|, \quad (2.50)$$

in which the penalty increment coincides with difference of labeling of two neighboring pixels. While convex constraints like this can be efficiently solved using some optimization tools, however they will results in poor surface boundaries, thus it is not “discontinuity-preserving”. The reason behind it is simple, depths change dramatically along depth surface boundaries, since the penalty coincides the increment of difference, it will result in over-smoothing the differences.

The most simple edge-preserving smoothness term is defined as

$$E_{smooth} = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) = \sum_{\{p,q\} \in N} \lambda T(f_p, f_q), \quad (2.51)$$

where

$$T(f_p, f_q) = \begin{cases} 0 & \text{if } f_p = f_q, \\ 1 & \text{otherwise,} \end{cases} \quad (2.52)$$

and this is usually called the Potts model.

Another simple term which also features edge-preserving is

$$E_{smooth} = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) = \sum_{\{p,q\} \in N} \lambda \min(|f_p - f_q|, \gamma), \quad (2.53)$$

where γ is a truncation threshold. It truncates the cost in case the depth changes dramatically.

Unfortunately minimizing such functions is NP-hard, therefore researchers have adopted several approximation methods for global optimization including iterated conditional models, graph cut, max-product loopy belief propagation, tree-reweighted message passing and so on. These algorithms have been briefly reviewed in section 2.2.

Chapter 3

Evaluation of Different Segmentation Algorithms and Their Performance in Stereo Matching

3.1 Overview of Image Segmentation and its Evaluation

The tasks in computer vision are often associated with the goal to find what objects or surfaces are presented in the scene. In this process, pixel based analysis usually lacks the capability of representing objects, therefore image segmentation based representation has been playing the crucial role instead. It divides an image into visually meaningful partitions and extracts the corresponding visual features of interest. Generally, segmentation has been applied widely in low-level vision applications such as image understanding, classification, stereo matching and others.

In the past decade, the development of segmentation algorithms has attracted significant attention and many approaches have been developed, meanwhile relatively few attention has been paid on their evaluations. Although most of the algorithms compare its result with some particular chosen algorithms in their papers, the comparisons are neither complete nor systematical, not even mention the great diversity of definition of “visually meaningful” segments [3], ranging from simple uniform intensity and color, homogenous textures, symmetric pat-

terns and up to complex semantically meaningful objects. Therefore, it is difficult for researchers to choose the one that most suit their application.

In this chapter, we will briefly review five state-of-the-art image segmentation algorithms [16] [21] [43] [49] [60](Figure 3.1), and evaluate their performances in stereo matching with multi-scale segments. To organize this chapter, section 2 describes the methodology of these five algorithms. In section 3, we will present the efficient evaluation framework. In section 4, we will test its performance in standard segmentation-based stereo matching by both qualitative and quantitative analysis. Finally, the conclusion are given in the section 5.

3.2 Five Modern Segmentation Algorithms

There are a great variety of segmentation algorithms. In the chapter, we will briefly review five selected state-of-the-art approaches.

Efficient Graph-Based Image Segmentation

Efficient graph-based image segmentation(*EGIS*)[21] defines a predicate by measuring the evidence for a boundary between two regions using a graph-based representation of the image. *EGIS* proved to be efficient by using two different kinds of local neighborhoods in constructing the graph. Moreover, it makes greedy decisions to produce segmentations that satisfy global properties. The algorithm runs in times nearly linear in the number of graph edges and is also fast in practice. It can preserve the details in low-variability image regions while ignoring in other high-variability regions. The global aspects of the image is well reflected by perceptually capturing the important groupings or regions.

Turbo Superpixels

Turbo superpixels [43] is a geometric-flow based algorithm for computing dense over-segments of an image. This approach not only respects local image boundaries but also limits under-segmentation through a compactness constraint. It is very fast and the complexity is approximately linear in image dimension, which reducing superpixel computation to an efficiently-solvable geometric flow problem. It yields less under-segmentation than other algorithms lacking of a compactness constraint, while offering a significant speed-up over N-cuts which does enforce compactness.

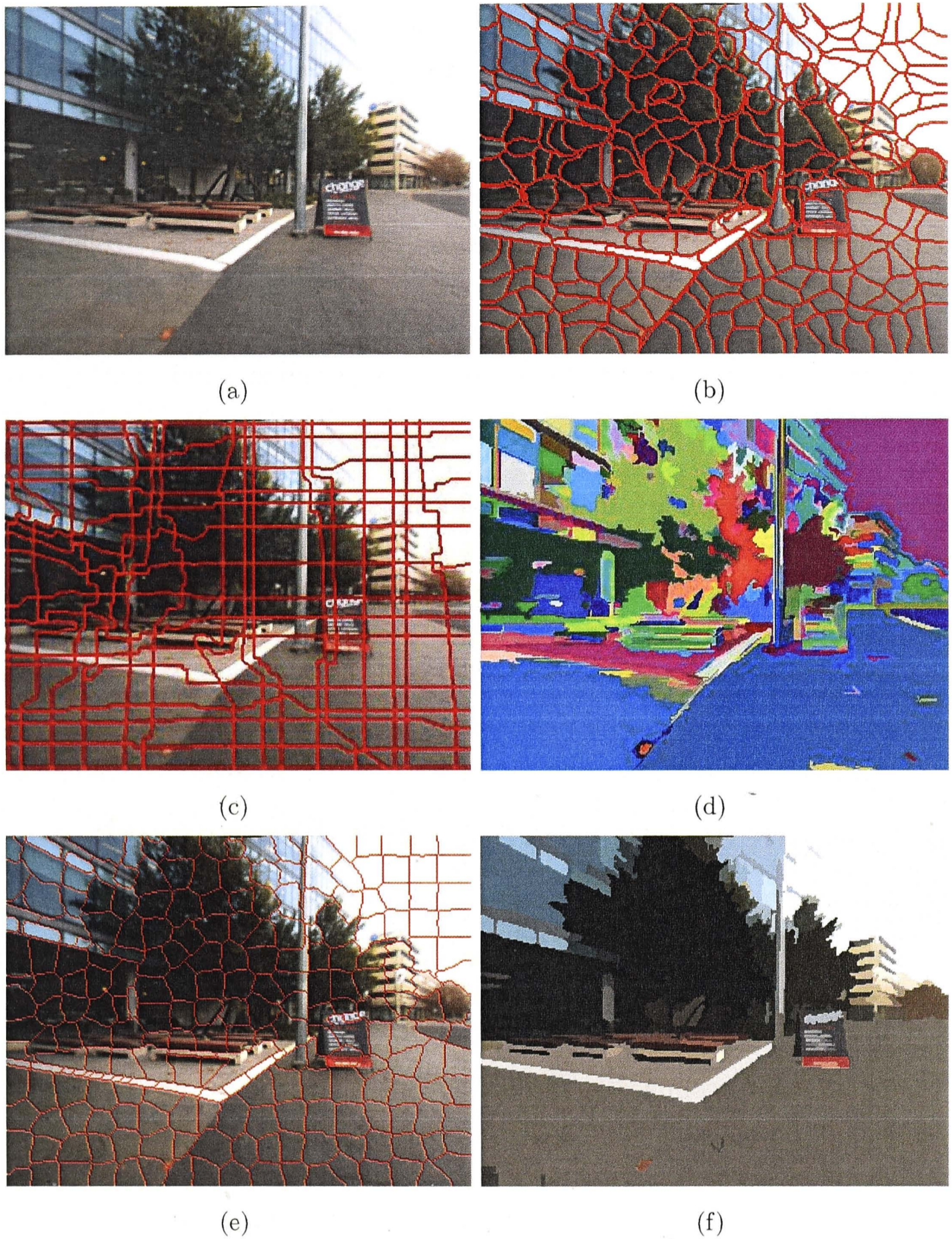


Figure 3.1: Image segmentation results by 5 different approaches: (a) Our captured outdoor image; (b) Superpixel result; (c) Superlattices result; (d) Efficient graph-based image segmentation result; (e) Turbo superpixels result; (f) Mean-shift result.

Superpixel Lattices

Superpixel lattices[49] is a method that produces superpixels which can preserve a regular topology of original pixels. Such topology is quite useful especial in

high order cliques related labels. The n^{th} superpixel has consistent position or relationship with its neighbors so it is easy to get the label of neighbors. It is also very fast and accurate.

Superpixel

Superpixel[60] is an over-segment method. It is a local, spatially-coherent, homogeneous, structure which preserves information over scales or sampling resolutions. In general, a superpixel can represent the property of the pixels in its region. Also it is easier to optimize as the number of nodes significantly decrease.

MeanShift

Mean-shift approach[16] is essentially defined as a gradient ascent search for maxima in a density function defined over a high dimensional feature space. The feature space includes a combination of the spatial coordinates and all its associated attributes which are considered during the analysis. The main advantage of the mean-shift approach is based on the fact that edge information is incorporated as well.

3.3 Segmentation Evaluation Framework

A few comprehensive reviews [84][83] on evaluation methods for image segmentation has been done. Generally, existing evaluation methods can be classified into two categories: analytical methods and empirical methods. The analytical methods analyzes and evaluates the algorithms themselves directly based on the principles, requirements, complexity and so on, while the empirical methods give their assessment by measuring the quality of segments. Moreover, according to [84], the empirical methods could be further divided into two sub-categories: goodness methods and discrepancy methods. In the former, the segments are measured by values of goodness on pre-defined evaluation systems, while the latter one is to compare the generated segments with the reference image(ground-truth in some sense), and the difference is quantitative measured.

3.3.1 Analytical Methods

The analytical methods directly assess the mechanism and properties of segmentation algorithms itself. The advantage of these algorithms is that it skips the actual implementation of the segmentation algorithms thus avoiding the differ-

ence in efficiency when implementation environment is not consistent. However, such properties are often hard to obtain or difficult to analysis. And the analysis results are not always objective and quantitative, for instance, some researchers attempt to evaluate the prior assumption that a segmentation algorithms use[14], and determine the goodness of the algorithm by judging the reasonableness of the incorporated prior. Generally, the development of analytical methods is limited, and most of the existing works are only associated with some specific models or desirable properties.

3.3.2 Empirical Discrepancy Methods

The empirical discrepancy methods determine the goodness of a segmentation by comparing the disparity between the segmented image by this algorithm and some reference image. The reference image is sometimes called the groundtruth. When the input image is manually synthesised, the reference image can also be easily obtained. But when the input image is a natural image, usually the human labeled segmentation is referred as the reference image. The commonly used discrepancy measurement is the mean-square signal-to-noise ratio as in [26]. In this case, a lower disparity value indicates a higher similarity and a better segmentation. In addition, several other discrepancy measures have been proposed as well. These measurements can be sorted into five categories: discrepancy based on the number of mis-segmented pixels, discrepancy based on the position of mis-segmented pixels, discrepancy based on the number of objects in the image, discrepancy based on the feature values of segmented objects and discrepancy based on miscellaneous quantities. For more details, please refer to [84].

3.3.3 Empirical Goodness Methods

At present, most methods to evaluate the quality of segmentation measurements are established according to the “ideal” segmentation of human intuition. Widely used evaluation ways include intra-region similarity, inter-region dissimilarity and region shape parameter. We adopted these three evaluations to multi-scale over-segments from these five approaches described in the previous section.

Intra-region similarity

The elements in a region should be similar. They include similar brightness, texture, and low contour energy inside the region. The homogeneous degree of

the features inside a region could be computed by the variance of the pixels inside [42].

$$\sigma = \frac{1}{N_i} \sum_{(x,y) \in R_i} [f(x,y) - \frac{1}{N_i} \sum_{(x,y) \in R_i} f(x,y)]^2 \quad (3.1)$$

where N_i is the pixel number inside R_i , $f(x,y)$ is the feature of pixel located at (x,y) . Because every image has different number of regions, this result should be normalized. Sahoo et al [56] proposed a normalized uniformity measure. We improve it as

$$\sigma_{nor} = 1 - (\bar{\sigma} - min)/(max - min), \quad (3.2)$$

in which $\bar{\sigma}$ denotes the average of all σ in the image. min is the minimal σ while max is the maximal in the images.

Inter-region dissimilarity

The ideal segmentation is to distinguish each region, and the elements between different regions are dissimilar. It means dissimilar brightness or texture, and high contour energy on region boundaries. Such properties may also be used to evaluate the segments. A good segmentation should divide a image into regions with higher contrast.

$$\sigma = \frac{|f_o - f_b|}{f_o + f_b} \quad (3.3)$$

f_o is the average of visual features in a foreground over-segments, while f_b is the average in the remaining regions as the background. The maximal mean of c between the foreground and the background represents the best segmentation.

Region shape parameter

Another method to evaluate the over-segments is region shape parameter. Different threshold can affect the extraction of the object boundary. We can define a parameter s which is closely related to the boundary as the boundary best representing the object shape. The parameter can evaluate the segmentation from the view of shape.

$$\sigma = \frac{1}{c} \left\{ \sum_{(x,y)} Sgn[f(x,y) - f_{N(x,y)}]g(x,y)Sgn[f(x,y) - T] \right\}, \quad (3.4)$$

where $g(x, y)$ is a gradient value at (x, y) , T is the threshold value selected for segmentation, c is a normalization factor and $Sgn()$ is the unit step function.

3.4 Evaluation of Performance in Stereo Matching

Here we will introduce our evaluation system of the performance of different segmentation method in stereo matching. The introduction of stereo matching can be found in Chapter 2. In general, most of the existing methods can be sorted into two categories: pixel-based and segment-based. Pixel-based algorithms are more easily to be disturbed by local noises. As human visual system partitioning the scene into a set of regions each of which has the same or similar visual features, segmentation has been widely used in the majority of modern stereo matching algorithms. In the past decade, a variety of segment-based stereo matching methods have emerged. These methods perform well in reducing the ambiguity associated with textureless regions and enhancing noise tolerance. However, researchers usually choose specific segmentation algorithm without comparing the different results using different segmentation methods. To establish our evaluation benchmark, we will briefly introduce our stereo matching algorithm.

Here we adopted a classic segmentation-based stereo matching algorithm, the algorithm is under the global framework. Unlike local methods emphasizes on localized matching cost computation and then simply assign the disparity label with the minimum cost value(usually referred to “winner-take-all” strategy), global algorithms prefer to seek a disparity assignment that minimizes a global cost function that combines both data term and smoothness term. The data term is usually directly defined as the pixel-based local matching score by shifting a predefined window along the horizontal directions between the candidates in left and right frames, and the smoothness term encodes the smoothness between the spatially neighbors. Once the energy function is defined, several algorithms can be used for energy minimization. In our framework, we choose graph cuts[11][10] algorithm which proved to be a proper tradeoff between the efficiency and the performance. To better encode the influence of different segmentation methods, the work is carried out on segmentation level. In our stereo matching framework, segment is treated as the minimal element and variable, and then as one individual node in graph cuts optimization. The spatially connection is also transformed from pixel-based 4-connected or 8-connected neighbors to the segmentation-based

neighboring system. An illustration of this is given in Figure. 3.2.

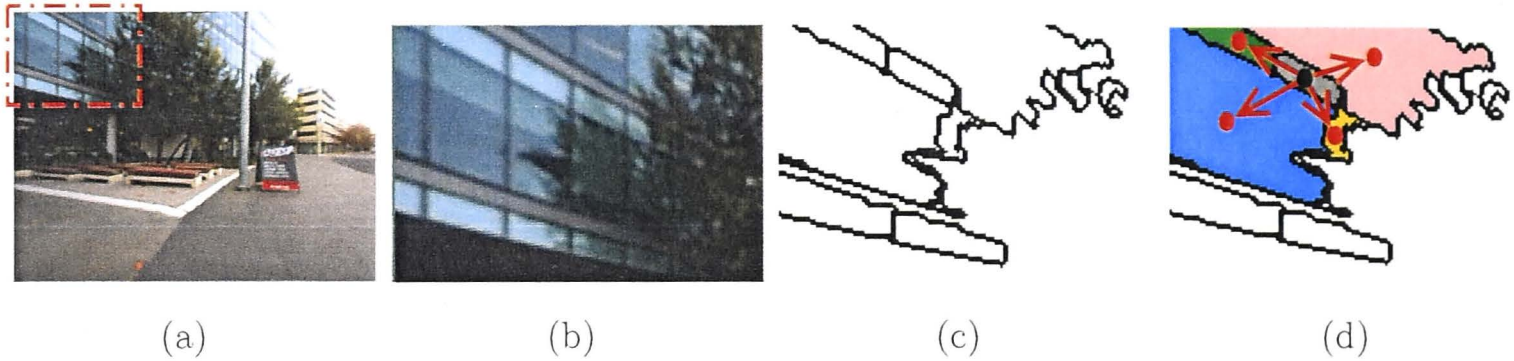


Figure 3.2: Segment-based neighboring system. (a) Our captured outdoor image; (b) Cropped image; (c) Segmentation result, where each white region is a segment; (d) Segment typology illustration, where the red nodes are the neighbors of the black node.

Let $X = x_1, x_2, \dots, x_n$ be the set of pixels, $C = c_1, c_2, \dots, c_n$ be the set of segments (cliques) and $L = \{1, 2, \dots, n\}$ be a set of n discrete depth value labels. The task is to find a labeling configuration f that allocates the labels from L to each variables $c_i \in C$. Each possible labeling f has its own *posterior* probability, the goal is to find the f^* that maximize the probability. According to the Hammersley-Clifford theorem, maximum a *posterior* labeling f^* (MAP) is equivalent to finding the minimum of the Gibbs energy. We define the proposed energy function as:

$$E = \underbrace{E(c_i)}_{DataTerm} + \underbrace{E(c_i, c_j)}_{SmoothnessTerm} \quad (3.5)$$

Data Term

The data term measures the cost of assigning a disparity to a certain pixel on the image. Data terms are often formed as the cost volume of pixel-based local matching such as *SSD*, *SAD* or *NCC*. Here we define ours according to [7]: let I_L and I_R be the left and right image respectively, Y is the corresponding pixels in the right image and \hat{I}_R is the linearly interpolated function of between the sample points of the right scanline, then the possibility that y_i matches x_i is defined as:

$$\bar{d}(x_i, y_i, I_L, I_R) = \min_{y_i - \frac{1}{2} \leq y \leq y_i + \frac{1}{2}} |I_L(x_i) - \hat{I}_R(y)|. \quad (3.6)$$

Symmetrically,

$$\bar{d}(y_i, x_i, I_R, I_L) = \min_{x_i - \frac{1}{2} \leq x \leq x_i + \frac{1}{2}} |I_R(y_i) - \hat{I}_L(x)|. \quad (3.7)$$

Then, the dissimilarity is defined as the minimum of these two:

$$d(x_i, y_i) = \min\{\bar{d}(x_i, y_i, I_L, I_R), \bar{d}(y_i, x_i, I_R, I_L)\}. \quad (3.8)$$

This design has proven to be insensitive to sampling error. Finally, the data term is computed as:

$$E(c_i) = \sum_{c_i \in C} \sum_{x_i \in c_i} d(x_i, y_i). \quad (3.9)$$

Smoothness Term

The smoothness term encourages neighboring segments to have similar disparity label which leads to a more smoothed disparity map and on some level eliminate minor mistakes caused by the local stereo matching. We exploit the form of Potts model, and only penalize on the difference.:

$$\phi(c_i, c_j) = \begin{cases} 1, & \text{if } c_i = c_j, \\ 0, & \text{otherwise.} \end{cases} \quad (3.10)$$

And the smoothness term is the sum of smoothness term over all pairs.

$$E(c_i, c_j) = \lambda \sum_{\{c_i, c_j\} \in C} \phi(c_i, c_j). \quad (3.11)$$

In the function, λ is the weight parameter balancing the scale between data term and smoothness term based on the scale of the segmentation. In our experiment, λ is empirically set for every image pair and kept the same during the implementation of all 5 segmentations.

Once the energy potential has been defined, we apply the powerful α -expansion of graph cuts to minimize the energy iteratively. We start with an arbitrary labeling f . In each iteration, one random label from L is taken out as the α , and for the nodes with current label other than α will be involved in this *alpha*-expansion by adding it to the graph. After the graph has been settled, a st-min cut is executed, and the labels of nodes are determined simultaneously. Saying the new labeling is f' , we compare its energy with the one from the last iteration, and if the difference is within a certain threshold for a certain number of times, then the optimization is terminated, and the current f' is set as the optimal labeling f^* .

To test different segmentation algorithms under different scales, we apply them on the same datasets (some are with groundtruth for quantitative analysis purpose) and all the number of segments are controlled to be similar. The three

scales are dividing an image into 100, 500 and 1000 segments respectively. Of course, not all the codes of algorithms allow us to predefine the proposed number of segments precisely, so the numbers are only be roughly determined. In our evaluation, the number of segments are all enforced to be within 5% offset of the designed number. For quantitative analysis, the groundtruth of the disparity map, occlusion map and depth discontinuity map are used. The quantity evaluation includes accuracy for non-occluded regions, all regions and depth discontinuity regions. For pixels, the absolute difference of their depth with ground truth are computed. Pixel with difference large than 1.0 will be labeled as a *bad* pixel. The error rate is the average percentage of these *bad* pixels.

3.5 Experiment

The evaluation has been carried out on Middlebury’s benchmark images[67](Venus, Teddy, Tsukuba, Cones) and our real-scene dataset. The real-scene data set we use is composed of seven outdoor and five indoor images which captured by Bumblebee stereo camera in our office and surrounding areas. The calibration and epipolar rectifying work have been done by Bumblebee itself. The testbed is on a desktop computer with Intel core I3 2.93Ghz CPU. It is worth noting that although most of the segmentation algorithms taking less than a few seconds to process an image with resolution of $640 * 480$ disregard of the three scales, but the superpixel algorithm may cost significant more time when the number of segments is risen. For example, it takes 5.25 minutes to segment an image into 1000 segments.

3.5.1 Empirical Goodness Evaluation

For empirical goodness evaluation of over-segments generated by these five approaches, we use all the 16 images and compute their average as the final result. Also, each image is segmented under three different scales: 100, 500 and 1000 samples, namely large, medium and small scales. Large scale means that the dimension of the over-segment is larger and the number of over-segments in a image is less, vice versa. For all three evaluations, the higher value in the table means the better quality of segmentation the approach obtains. The result is in Table. 3.1.

To have it more clearly presented, the average results over three scales are shown in Figure. 3.3.

		Intra-region	Inter-region	Region Shape
SuperPixel	Large Scale	0.7148	0.1724	0.2049
	Medium Scale	0.8784	0.1746	0.1848
	Small Scale	0.8080	0.2034	0.1720
SuperLattice	Large Scale	0.7480	0.1749	0.2377
	Medium Scale	0.8044	0.1904	0.2042
	Small Scale	0.8102	0.1920	0.1796
MeanShift	Large Scale	0.6516	0.2035	0.3747
	Medium Scale	0.7842	0.1641	0.2615
	Small Scale	0.7129	0.1935	0.2394
EGIS	Large Scale	0.8386	0.4003	0.4438
	Medium Scale	0.8365	0.2420	0.3891
	Small Scale	0.8163	0.2737	0.3278
TurboSuperpixel	Large Scale	0.7233	0.1996	0.2305
	Medium Scale	0.7998	0.1373	0.2436
	Small Scale	0.7855	0.1785	0.2279

Table 3.1: Quantitative analysis on empirical goodness measurement over all three scales.

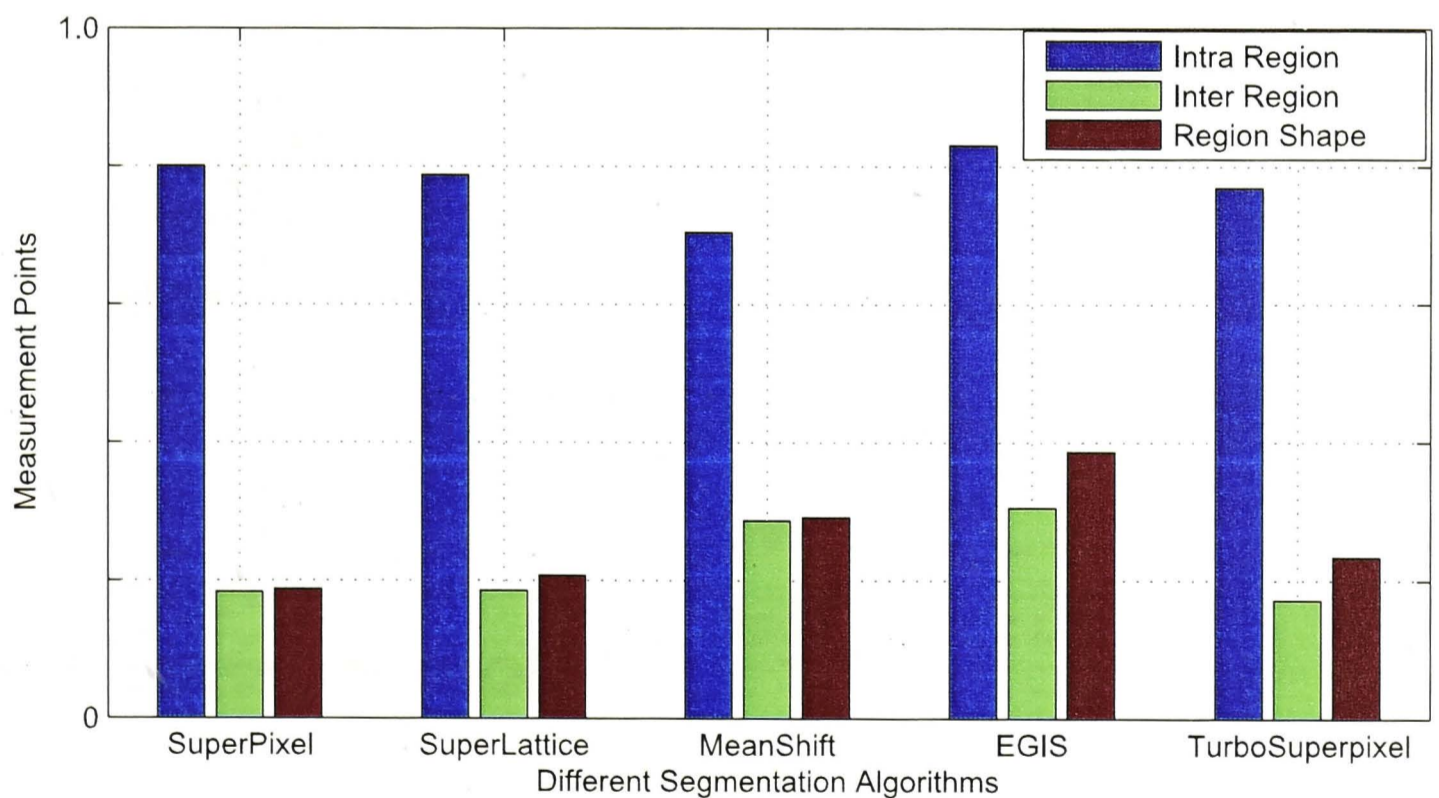


Figure 3.3: The average empirical goodness evaluation results of three scales.

From Table. 3.1 and Figure. 3.3, it explicitly shows that *EGIS* obtained the best performance under the evaluation of intra-region similarity, as it got the maxima of the average with the value of 0.8305, and also there is a consis-

tency of similarity among all three scales of EGIS. In inter-region dissimilarity measurement, EGIS also obtained the highest score at the average(0.3053). In the measurement of region shape, along the scale decreasing, the value of shape parameter consistently decreased on all of five approaches. This phenomenon can be explained that large-scale over-segment is more coincided with objects than small scale's, and gradients on boundaries are significantly greater. In general, *EGIS* and MeanShift give the best performance over other three algorithms.

3.5.2 Performance Evaluation in Segment-based Stereo Matching

To test the performances of different segmentation algorithms, all 12 image pairs are used. In terms of efficiency, it takes less than 40 seconds to process an image pairs in average. The accuracy has been calculated with the average of four image pairs from Middlebury benchmark, namely Teddy, Tsukuba, Venus and Cones. The error rate for non-occluded regions, all regions and depth discontinuity regions are shown in Figure3.4, Figure3.5 and Figure3.6 respectively. For all three figures, the lower the value is, the better the depth is.

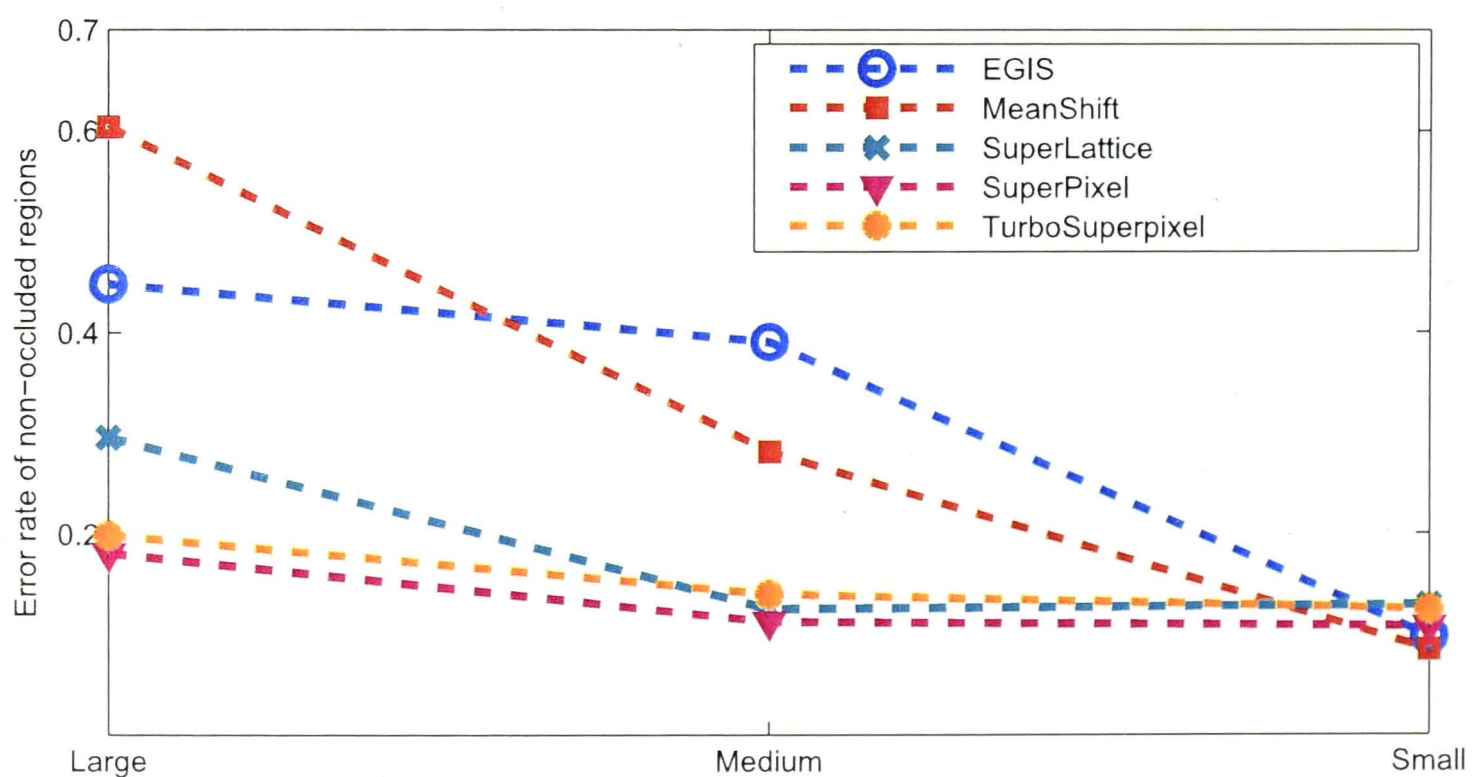


Figure 3.4: Error rate of non-occluded regions by 5 segmentations.

It clearly shows that segments from five algorithms contribute different effects on the disparity results. In general, the error rate decreases along with the scale

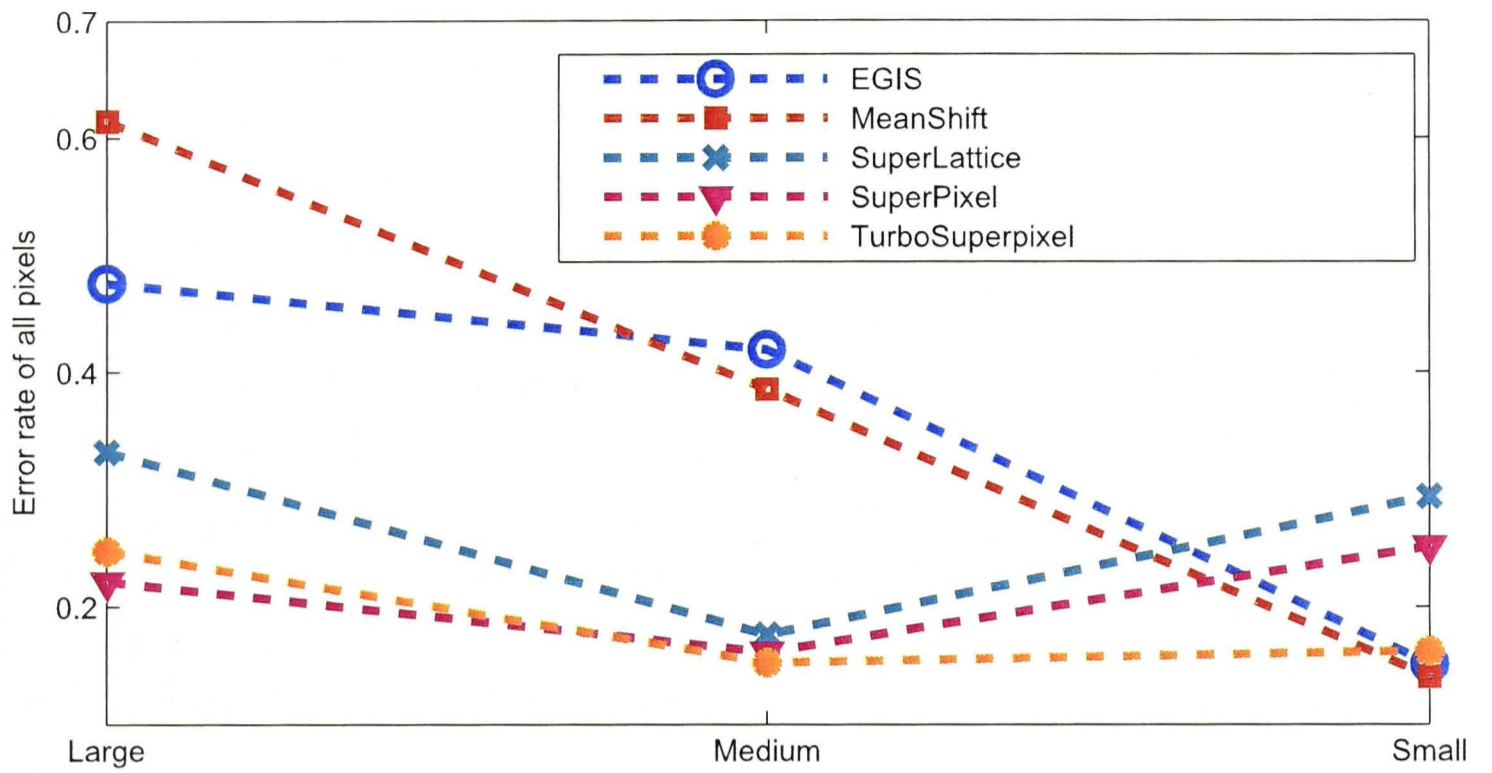


Figure 3.5: Error rate of all regions by 5 segmentations.

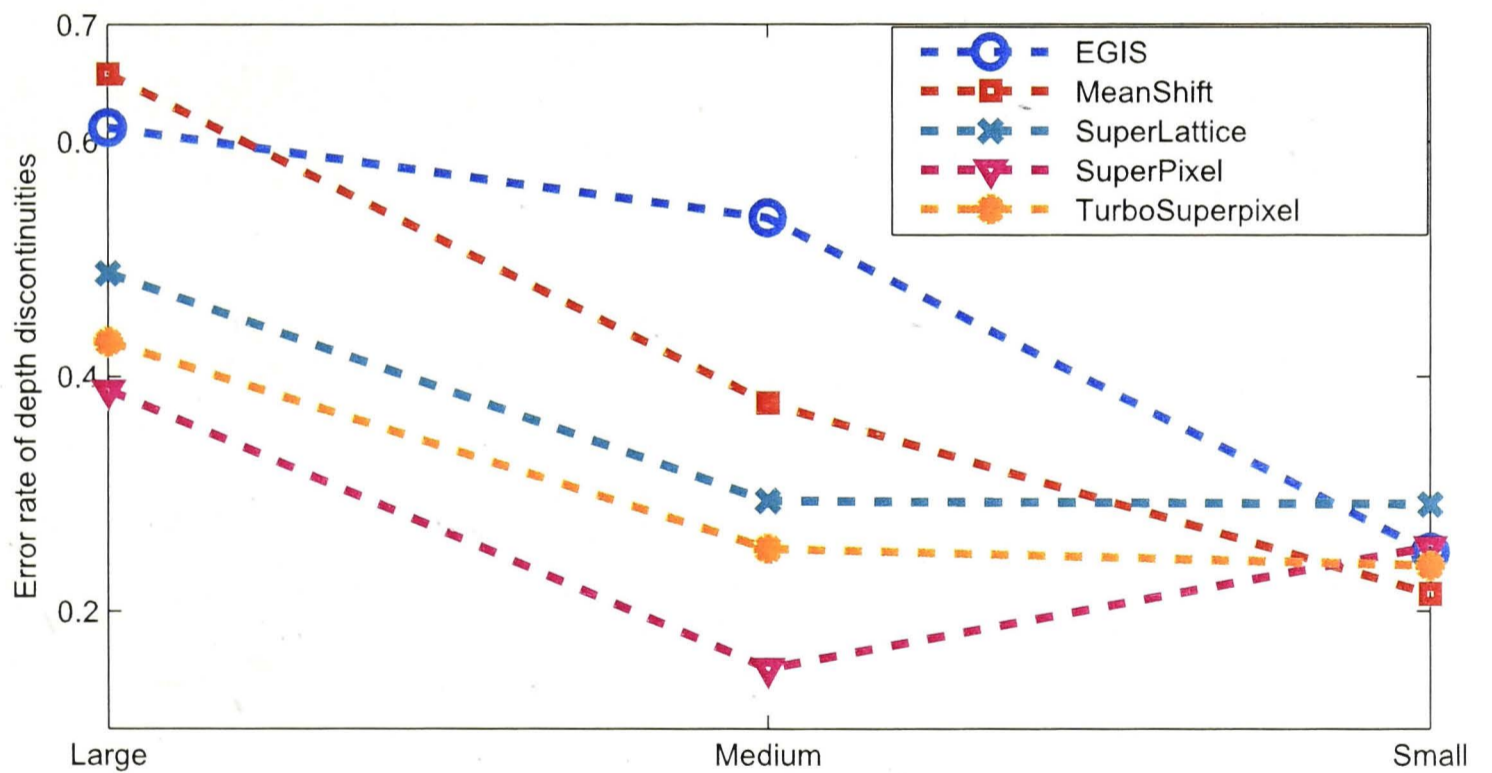


Figure 3.6: Error rate of depth discontinuity regions by 5 segmentations.

decreasing. Because in small segmentation scales, objects can be presented by a larger number of segmentations leading to a more accurate depth. In addition, MeanShift and *EGIS* give better performance when the scale is small, and when the scale is large, SuperPixel SuperLattice and TurboSuperpixel are more suitable when the scale is large. This is due to different natures of segmentations.

MeanShift and *EGIS* do not have constraint on segments' size and their color-based nature make them more coincident with the object boundaries or surface boundaries. This advantage make them suitable for pixel-based or small-scale segment-based stereo matching. On the other hand, SuperPixel SuperLattice and TurboSuperpixel have strong enforcement on segments' size, although it will bring artifacts into the depth, the regularization gives better performance when the scale is large, in which every segment becomes a large "pixel". Based on that, researchers proposed efficient stereo matching under low-resolution[70].

For quality review, we selected the Teddy image from Middlebury and one indoor and one outdoor images, as shown in Figure 3.7, Figure3.8 and Figure 3.9. The segmentation results are also presented for reference. Because the depth is enforced to be consistency inside each segment, the "blocky" phenomenon occurs on the results especially when the segmentation scale is large. Nevertheless, it gives a smoother depth distribution and filters isolated noises, as ground shown in Figure3.8 and wall shown in Figure 3.7. Over all, MeanShift outperformed the other four in the quality evaluation.

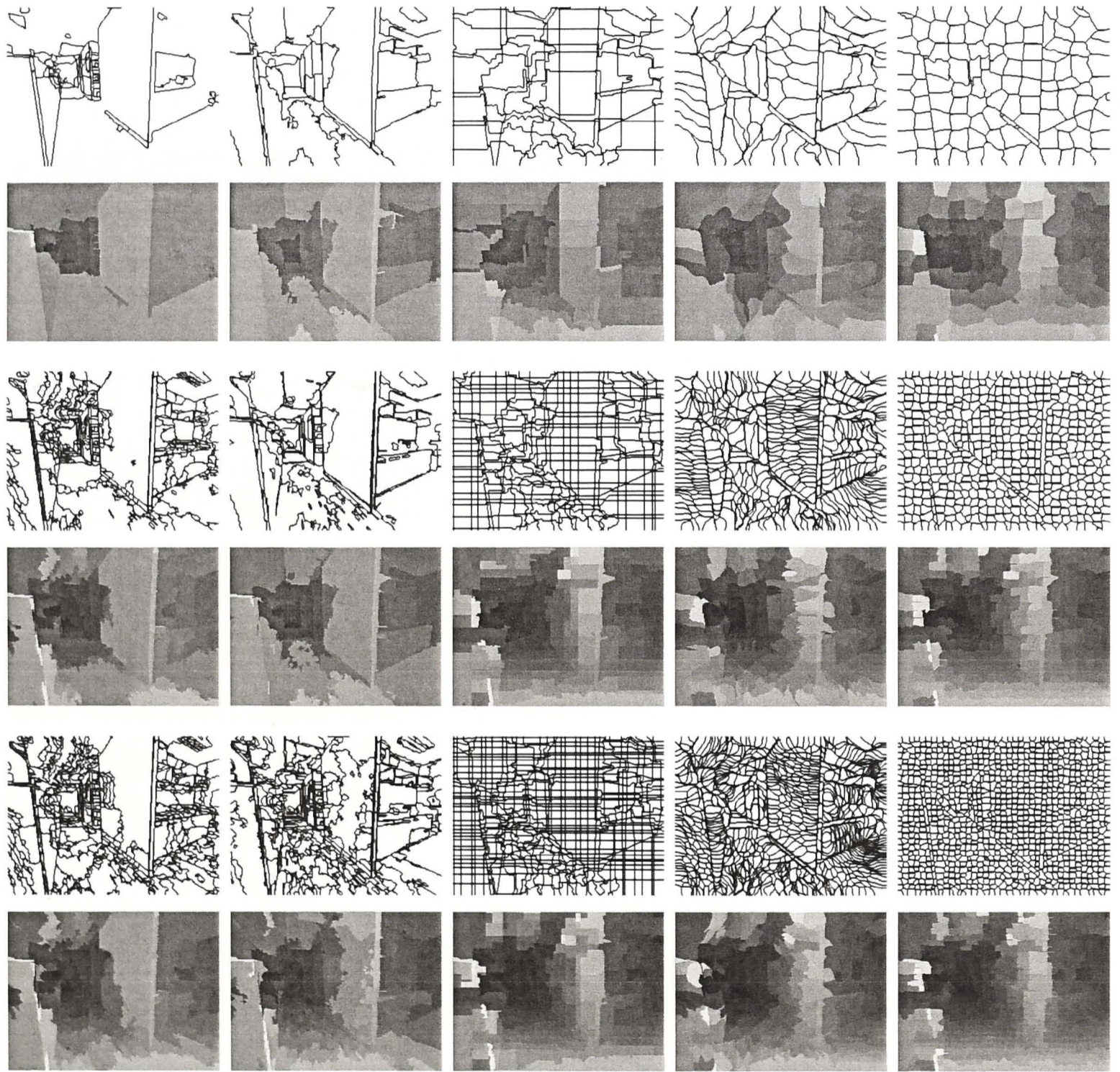


Figure 3.7: Performance evaluation on one indoor image pair. From left to right(step by 1 column): Meanshift, EGIS, SuperLattice, Superpixel, TurboSuperpixel, from up to down(step by 2 rows): Large scale, Medium scale and Small Scale.

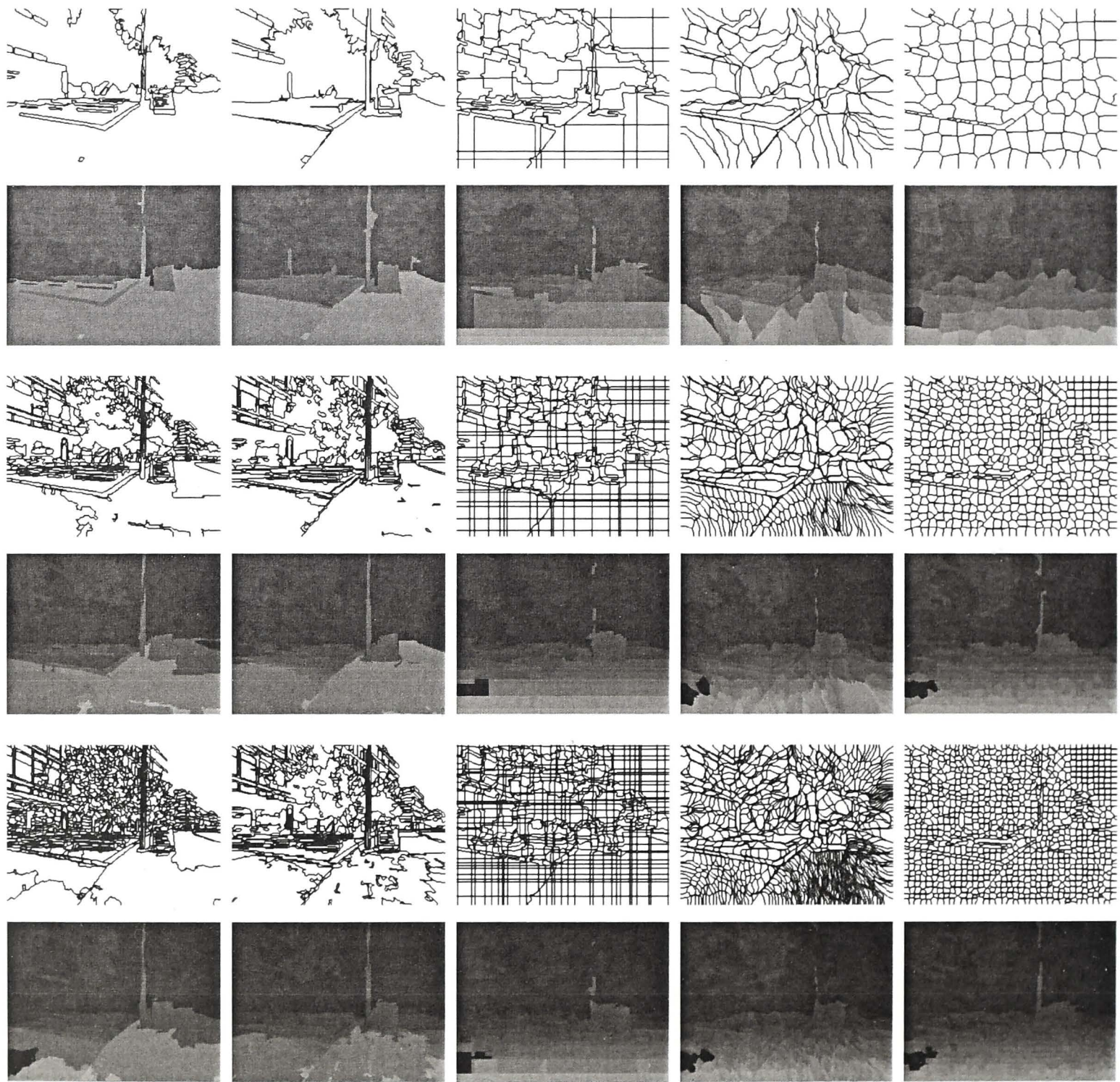


Figure 3.8: Performance evaluation on one outdoor image pair. From left to right(step by 1 column): Meanshift, EGIS, SuperLattice, Superpixel, TurboSuperpixel, from up to down(step by 2 rows): Large scale, Medium scale and Small Scale.

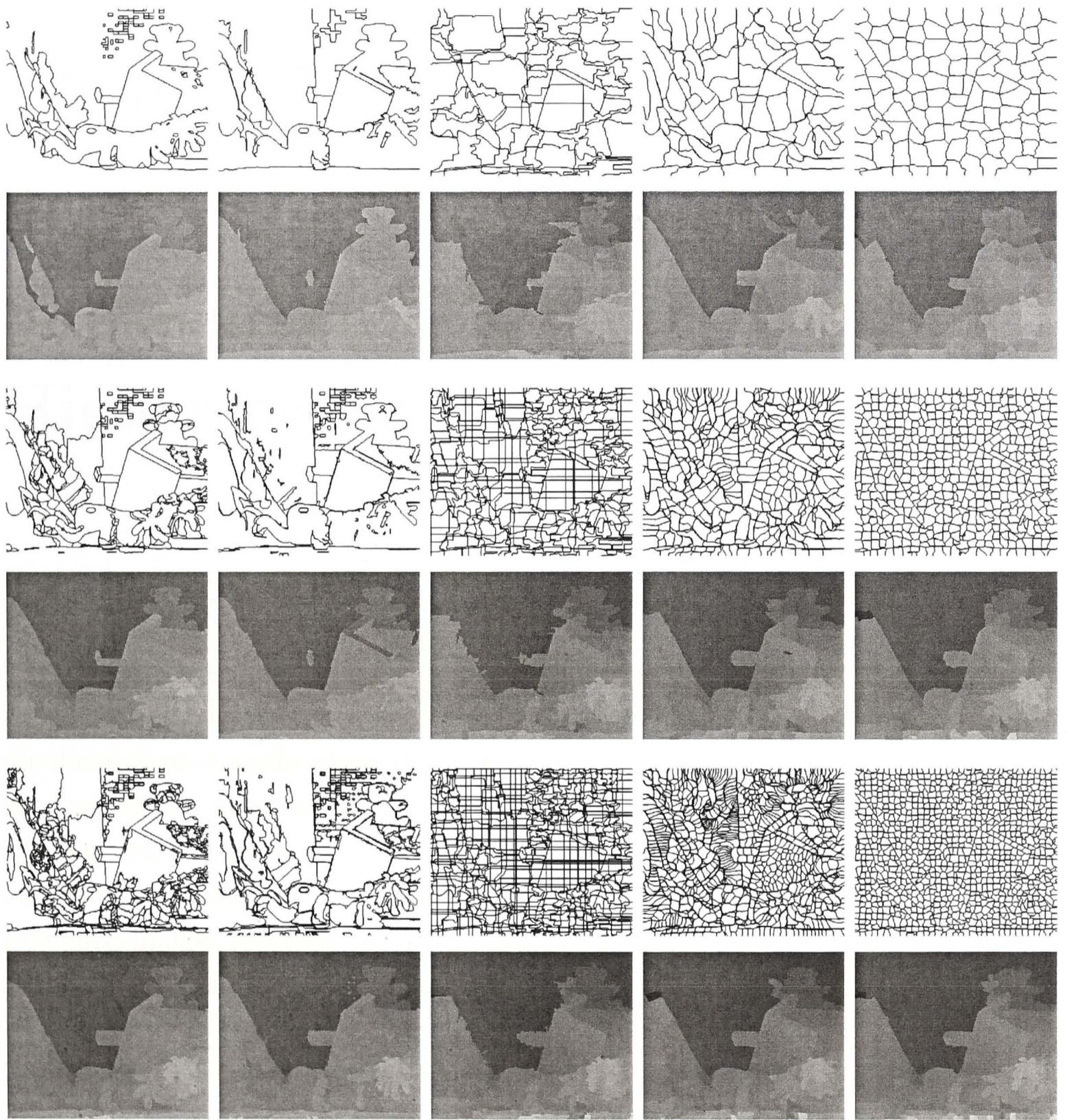


Figure 3.9: Performance evaluation on the Teddy image pair from Middlebury. From left to right(step by 1 column): Meanshift, EGIS, SuperLattice, Superpixel, TurboSuperpixel, from up to down(step by 2 rows): Large scale, Medium scale and Small Scale.

Chapter 4

Stereo Matching Using Sub-segmentation and Robust Higher-order Graph Cuts

4.1 Introduction

There exists a variety of color segmentation-based stereo matching algorithms which have shown accurate estimation of depth. Most of them usually share the *hard constraint* assumption that all pixels in the the same segment must have the same depth value or lie on a locally fitted surface such as a plane, and discontinuities only occur on segment boundaries[73][78]. A typical procedure of these algorithms is started by employing a visual feature based segmentation on the reference image, followed by a plane-fitting procedure based on the initial disparity estimation in each segmentation. Unary and smooth encouraging pairwise terms are then defined based on plane parameters within an energy minimization framework. Eventually global optimization algorithms are applied to solve the problem, like graph cuts[11] or belief propogation[20]. And it is worth noting that some methods generate possible plane proposals based on plane-fit, and use it directly as label set, and optimize it on segment based level. This should also be considered as *hard constraint*.

While *hard constraint* helps to reduce ambiguities of depth within textureless regions, it has several drawbacks. First, it is not robust. It purely relies on initial segmentation and local matching, and can not be recovered from noise and errors in these initial estimations. Second, there is no such simple relation between visual features and depth values, so clearly it is unreasonable to force pixels

inside one segment to lie on the same disparity plane.

Then is it possible to combine segmentation as a *soft constraint* into a pixel-based framework so that both of their advantages will be kept? In this chapter, we will present a novel framework, it does not force but encourage pixels to follow a certain disparity distribution if they are in the same segment. Also the *soft constraint* is realized in a higher-order term and being optimized under the same MRF model along with pixel based unary and pairwise potentials. We believe it is a more flexible and natural description of disparity distribution especially considering natural scenes.

Optimization for stereo matching is always a challenge. State-of-the-art local methods[81][61] are usually improved filter-based window matching. They maintain efficiency by avoiding global optimization procedure at the price of loss in quality and smoothness, and it obtains ambiguity results in textureless regions. Global methods on the other hand can involve more sophisticated assumptions and hence achieve better results. The problem with global methods are lack of efficient optimization tools. Message-passing based algorithms, such as efficient belief propagation(BP)[20] and tree-reweighted Tree-Reweighted Message Passing(TRW)[72] are often easily to be trapped in local-minimum or slow. Graph cuts is another powerful tool, but it has its own restrictions on the form of potential which commonly known as “submodularity” [38]. In some cases[9][48], the models are so complex that they do not satisfy submodularity any more. To solve this, QPBO-based optimization algorithm[63] is applied but only part of the nodes will be labeled. In our proposal, we take the form of the Robust P^n Potts model[35] which is proven to be submodular, therefore we can take advantage of the standard graph cuts for optimization which guarantees efficacy.

We also exploit the idea of sub-segmentation in our proposed method. Most existing paper directly use the result of segmentation as their stereo matching input. However, such the prior segmentations are only depend on visual features, and has no clear relationship with disparities. So we bring the relationship into a higher level, we further divide segments into disparity relevant sub-segments. And since this step is not always accurate, so we only define higher-order based *soft constraint* on it. It is worth noting that in [9], they also use the term *sub-segmentation*, but it is very different from ours. Firstly their definition of sub-segmentation are totally different from ours, secondly the only higher-order term in their model is the MDL term which penalize on the number of appeared labels and it is irrelevant to sub-segments, while we use sub-segmentation as the element of higher-order potential.

In addition, several known techniques are combined under the same framework, including symmetric occlusion handling, confidence measurement and plane-fitting.

The rest of the chapter is organized as follows. In section 2, we describe our algorithm in details. Experimental results are proposed in section 3.

4.2 Stereo Matching Through Robust Higher-order Graph Cuts

The main steps of our algorithms are illustrated in Figure 4.1. Generally it is a coarse to fine framework. First, MeanShift color segmentation[16] is applied to divide the reference image into several initial segments. Second, the fast and efficient Birchfield and Tomasi's pixel dissimilarity measure[7] constructs the correlation volumes for both left and right images respectively. And a winner-take-all strategy is applied afterwards. Third, we adopt the mutual consistency check(left-right cross-check) to classify the pixels into occluded and unoccluded pixels. Fourth, a confidence measurement is carried out on unoccluded pixels. A robust voting based plane fitting procedure is exploited on those chosen unoccluded pixels with high confidence to obtain an fitted disparity surface inside each segment, followed by novel sub-segmentation process. Finally, the robust higher-order graph cuts optimization is carried out to obtain the optimal result.

4.2.1 Initial Steps

Let X and Y be the sets of pixels of left image(I_L) and right image(I_R) respectively, L be the label set with n discrete depth values. The labeling problem is to find a labeling configuration f that allocates the labels from L to each variables $x_i \in X$.

First, MeanShift color segmentation[16] is applied to divide the reference image(here we define left image as reference image) into several initial segments.

In terms of cost volumes construction, an attempt of several local matching algorithms has been made in our experiment. Here we choose Birchfield and Tomasi's pixel dissimilarity measure[7] as our local based stereo matching approach, simply due to its better performance and its nature of being insensitive to sampling difference. The dissimilarity $c(x_i, y_i)$ is defined symmetrically as the minimum of two quantities which stand for how well the pixel in one image fits

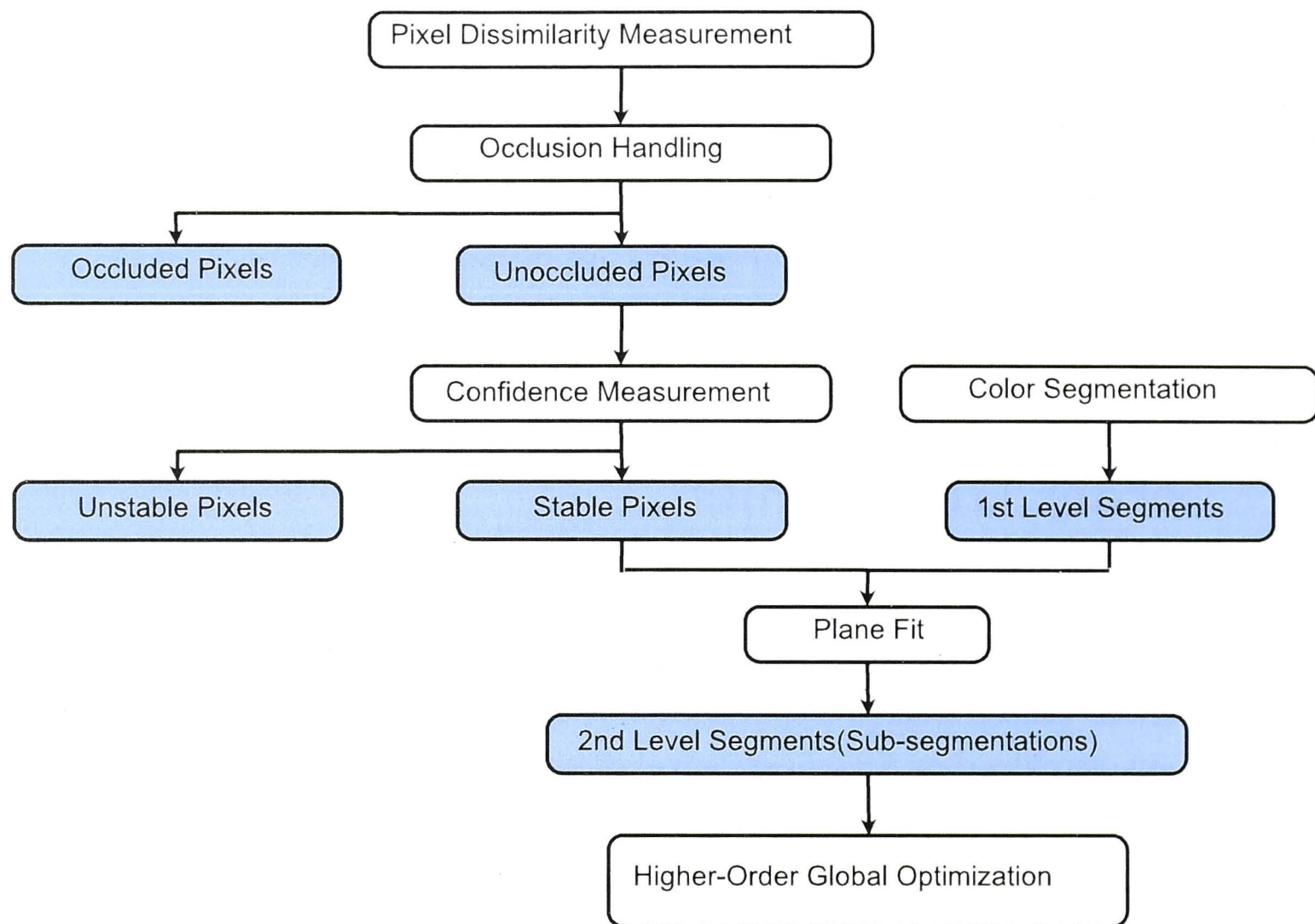


Figure 4.1: The flowchart of proposed algorithm.

into the linearly interpolated scanline surrounding the corresponding pixel in the other image, the details have been given in the previous chapter.

$$c(x_i, y_i) = \min\{\bar{c}(x_i, y_i, I_L, I_R), \bar{c}(y_i, x_i, I_R, I_L)\}. \quad (4.1)$$

Then a winner-take-all strategy is employed to select one label l_{x_i} with the minimum cost for every x_i among its multiple candidates. An example of the result is shown in Figure.4.2.

4.2.2 Occlusion Handling

Due to different geometries of the scene, it happens that some regions only appear in one of the images, and this phenomenon is commonly known as occlusion. Occlusion has always been a challenge in stereo matching, and researchers have proposed several algorithms to emphasize it. Generally, these algorithms can be sorted into five categories[19]: Bimodality Distribution(BD), Confidence Measurement Jumps(CMJ), Left-Right Cross Checking(LRCC), Ordering Requirement(OR) and Occlusion Constraint(OC). The first two algorithms belong to the

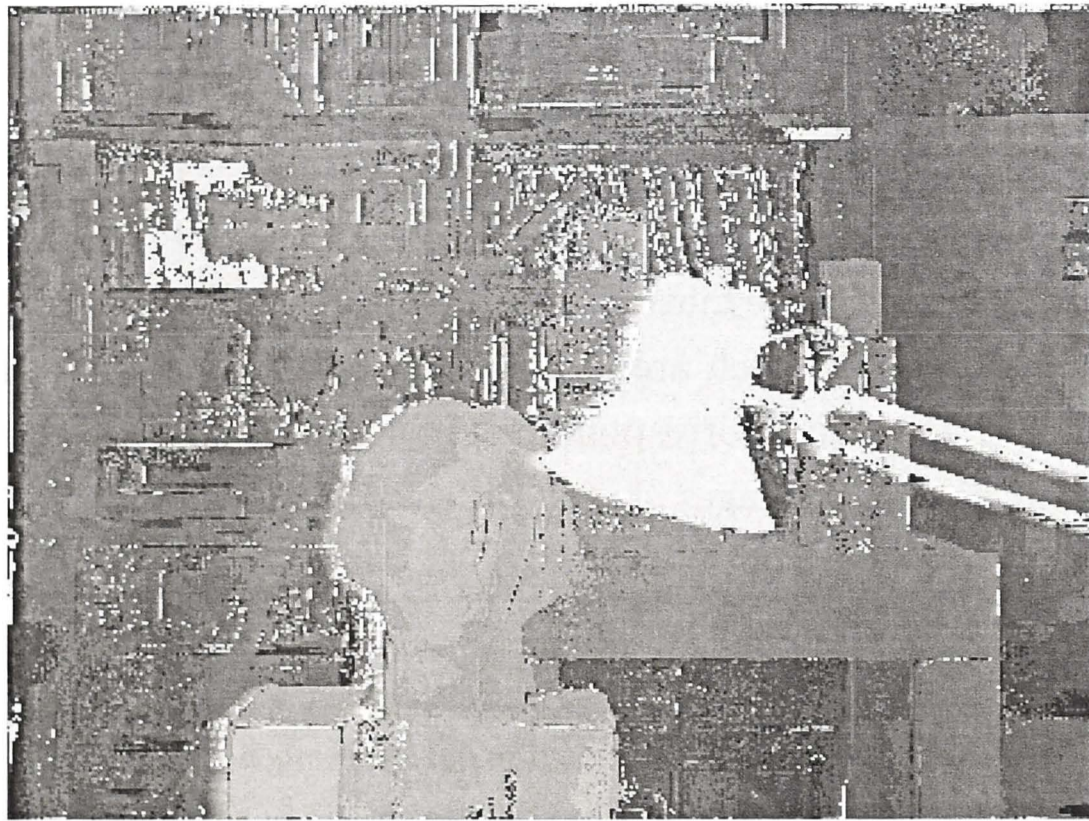


Figure 4.2: An example of the local matching result after locally winner-take-all.

border detection methods, while the last three focus on region detection.

Bimodality distribution(BD)

The theory of bimodality distribution is that half-occluded borders should have its neighbors in scanline arisen significantly, resulting in a bimodal distribution. Therefore, they usually compare the two peaks in the horizontal histogram:

$$\psi(BD) = \frac{\max(p_1)}{\max(p_2)}, \quad (4.2)$$

where $\max(p_1)$ and $\max(p_2)$ are two highest peaks. As the ratio $\psi(BD)$ is approaching to one, the center pixel is more likely to be on the occlusion borders.

Confidence Measurement Jumps(CMJ)

The fundamentals behind CMJ is simple, it measure at the goodness of the matching, and assumes that if the point in 3D world is visible in both views, then it should have been well matched. And for those pixels that only appear in one of the views, their confidences are supposed to be low. As a result, the occlusion borders are located in the places that the goodness measurement values jump the most. More formally,

$$\psi(CMJ) = \max(\bar{C}_x - \bar{C}_{x+w}, \bar{C}_x - \bar{C}_{x-w}), \quad (4.3)$$

where x is the horizontal coordinate of the center pixel, and \bar{C} is the sum of matching cost within a windows with size of w .

Left-Right Cross Checking(LRCC)

Left-right cross checking is the most commonly used hypothesis. Its basic assumption is that for the points which are presented in both views, their projections in both views should be mutual corresponding. In other words, corresponding pixels in left and right disparity images should differ only in occlusion areas. Analytically, let $dist(x_i)$ to be horizontal distance of x_i , and $proj(x_i)$ to be its projection in the other view, then we have

$$\psi(LRCC) = dist(proj(proj(x_i))) - dist(x_i). \quad (4.4)$$

For pixels with their $\psi(LRCC)$ other than 0 are failed in the mutual consistency check and labeled as occluded. However this occlusion detected by LRCC are not include occlusion regions but also textureless or false matched regions.

Ordering Requirement(OR)

The ordering requirement hypothesis is that every pixel corresponds to a unique point in the 3D scene, so for both views, the points that pixels correspond to should share the same ordering. Namely, let x_i and x_j be two pixels lie on the same scanline, without loss of generality assume x_i is on the left of x_j , then $proj(x_i)$ should also be located on the left of $proj(x_j)$ in the other view.

$$\psi(OR) = (dist(x_i) - dist(x_j)) * (dist(proj(x_i)) - dist(proj(x_j))) \quad (4.5)$$

Pixels with $\psi(OR) < 0$ will be labeled as occluded. Ordering requirement is known to fail in some cases, an example is given in Chapter 2.

Occlusion Constraint(OC)

The occlusion constraint assumes that disparity change smoothly within non-occluded surfaces. If depth changes dramatically in one view while jumps over pixels in the other view, then OC will label the unmatched pixels in the other view as occluded.

In our algorithm, we adopt the LRCC as our occlusion detection method, and an example of the result is presented in Figure4.3. After this step, pixels are labeled as either *Occluded* or *Unoccluded*.

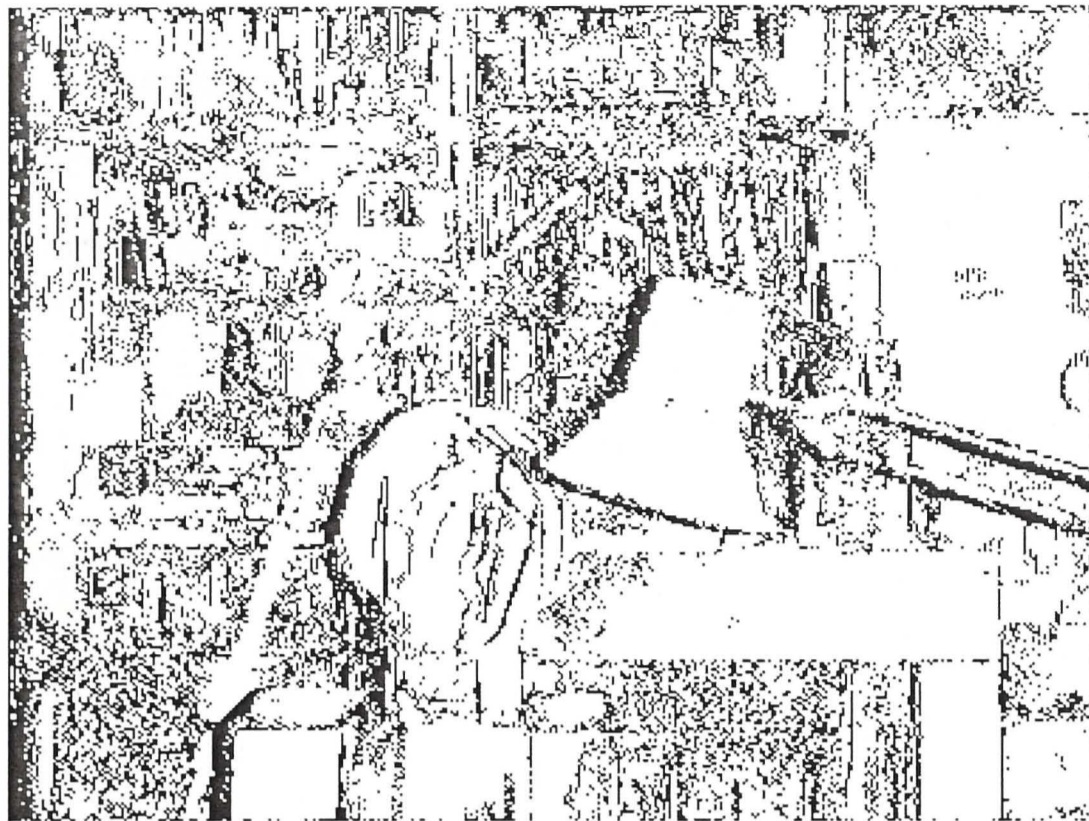


Figure 4.3: An example of our LRCC occlusion detection. In the figure, detected occluded pixels are in black, unoccluded pixels are in white. It can be seen that this detection is not accurate, it will not distinguish false matchings with real occlusions.

4.2.3 Confidence Measurement

Although winner-take-all strategy can easily assign the optimal depth label d for every pixel by choosing the minimum matching cost $c(x_i, d)$ among the candidates, the reliability behind the assignment may differ. For example, in textureless regions, the matching costs are intend to be quantitatively similar, so the optimal label may not be distinguished with confidence. A illustration of this phenomenon is given in Figure 4.4.

To solve it, similar to [77], we define our confidence function $conf(x_i)$ as:

$$conf(x_i) = \left(\sum_{d \neq d^*} \exp(-(c(x_i, d) - c(x_i, d^*))^2 / \sigma^2) \right)^{-1} \quad (4.6)$$

In the equation, $c(x_i, d)$ is the locally matching cost of pixel x_i at different depth candidates d , d^* is locally picked optimal label through winner-take-all process, σ is a scale robust parameter. In our experiment, a threshold θ is set for the generated confidence map. For *unoccluded* pixel x_i , if $conf(x_i) \geq \theta$, it is denoted as *stable* points, otherwise it is *unstable*. An example is presented in Figure 4.5.

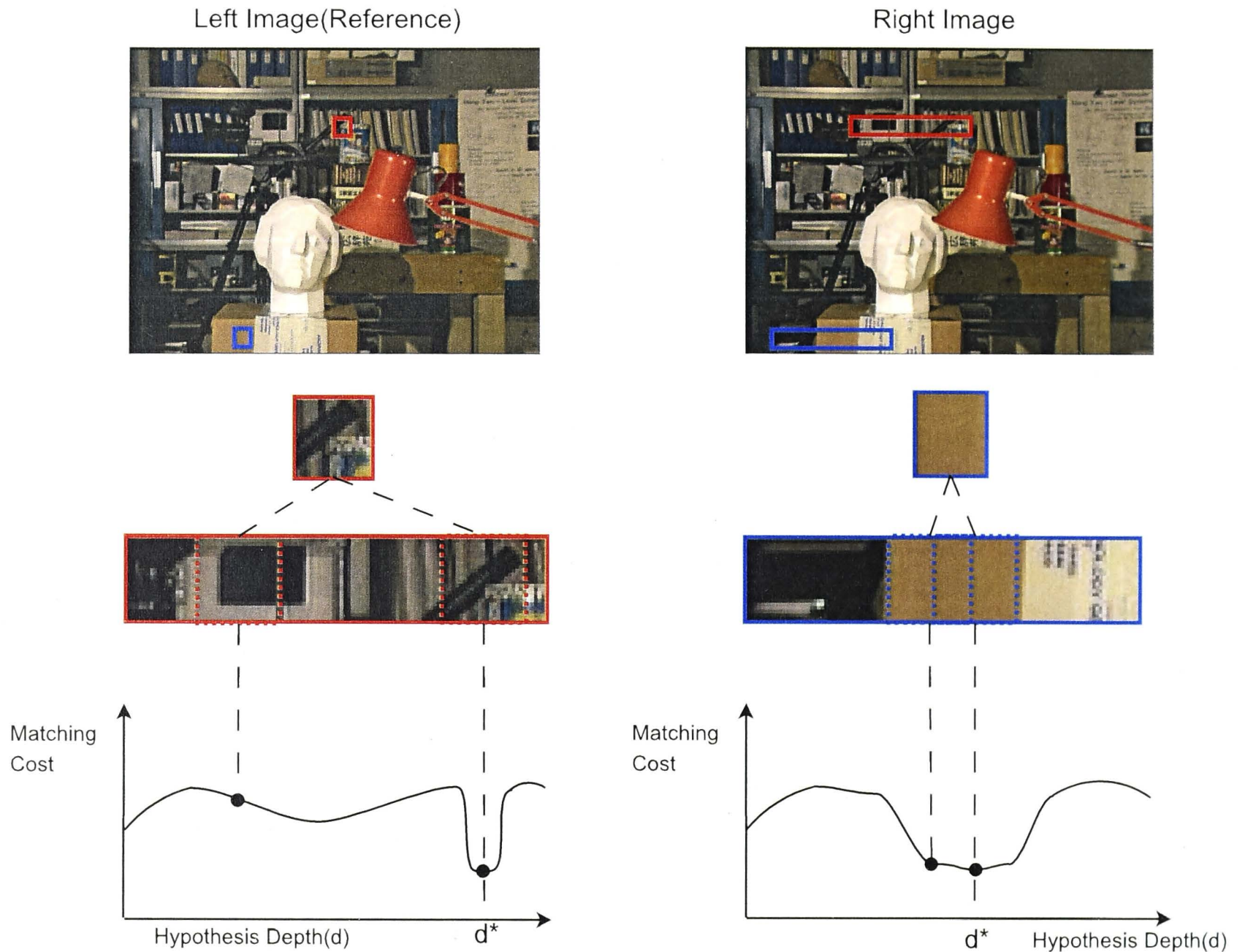


Figure 4.4: The fundamentals behind the confidence measurement. In high-texture regions(as shown in red block), the optimal(minimum) matching cost may significantly lower than other's, so the confidence should be higher. In contrast, in textureless regions(blue), the optimal matching cost may not differ that much comparing to others, so the confidence should be lower.

4.2.4 Plane Fitting

The most commonly used plane fitting techniques are Least Square Error(LSE)[76] and Random Sample Consensus(RANSAC)[22]. The advantage of the LSE is its efficiency, but its robustness is not comparable to RANSAC. Recently, a Voting-Based Method(VBM)[73] has been tested well in depth plane fitting.

Analytically, we establish a 3-dimensional space with x and y being the horizontal and vertical coordinates of the image and d being the depth value, then every $2d$ pixel $p_i(x_i, y_i)$ can be projected into a $3d$ point $p_i(x_i, y_i, d_i)$ once its depth



Figure 4.5: An example of our confidence measurement. Brighter the pixel is, the higher possibility of it being the right match it has.

value is known. Here we use the locally optimal depth map as the input d . Then the task is to find a plane $\hat{\beta} = [A; B; C]$ that fits these $3D$ points cloud, where A, B, C are plane parameters. Once the plane parameters are obtained, projections of each point on the plane can be computed through $d'_i = [x_i, y_i, 1] * [A; B; C]$. The error for fitting the pixel p_i to plane $\hat{\beta}$ is computed as the distance:

$$\Delta(p_i) = |[x_i, y_i, 1] * \hat{\beta} - d_i|. \quad (4.7)$$

The optimal plane $\hat{\beta}^*$ is the one that suits the $3d$ points cloud the most with respect to the inliers.

Least Square Error(LSE)

The LSE used here is the linear least squares[76]. By “least squares”, it means that the solution is approximated by minimizing the sum of the squares of the errors over all points:

$$sum = \sum_{p_i} \Delta(p_i)^2. \quad (4.8)$$

The optimal plane that minimizes the sum can be computed through

$$\hat{\beta}^* = (P^T P)^{-1} P^T D, \quad (4.9)$$

where P is the matrix of $2D$ points with the third coordinate as 1, and D is the matrix of corresponding input depth d .

Since the input depth map here is from locally matching, it inevitably includes lots of noises from false matchings. Although LSE is relatively efficient, it often can not give a robust estimation.

Random Sample Consensus(RANSAC)

RANSAC[22] is a robust method to plane-fitting. It dynamically divide the input data into two sets: inliers and outliers, and increasingly improve its estimations iteratively. In each iteration, it randomly samples 3 pixels, and generate a plane. Then it adds a process evaluating the number of inliers, and only proceeds when the model is regarded as qualified. Rather than give an estimation to fit all of the points, it is only applied on inlier points. The algorithm for RANSAC in our method has been given in Algorithm 1. The drawback of RANSAC is that several robustness relevant parameters have to be predefined. In addition, it is not efficiently comparable to LSE and voting-based method.

Voting-Based Method(VBM)

The idea behind VBM is simple. The proposed depth on the plane can be defined as $d'_i = [x_i, y_i, 1] * [A; B; C]$, so the plane parameter A can be obtained by calculating $\delta d' / \delta x$ for a pair of points along X -axis. By doing counting for every pairs, we can build a one-dimensional histogram with integer values of A as horizontal coordinate and count number as vertical coordinate. Once the histogram is made, by applying winners-take-all, value A is easily computed. After that, a similar strategy is applied on B by calculating $\delta d' / \delta y$. Once A and B are obtained, C can be settled by a similar voting operation. Unlike RANSAC, it is not required to predefine parameters, and its performance is comparable to RANSAC in most of the regions while at significantly higher efficiency. Its drawback is that it does not perform well in sub-pixel planes.

Here we perform a plane-fitting on the initial segments by Meanshift. Because plane-fitting plays an important role in our algorithm and further sub-segmentation will be based on it, algorithm insensitive to outliers is required. We have test all three algorithms in our experiment, the performances are shown in Figure 4.6. RANSAC achieves the highest accuracy, as a result, we choose it as our plane-fitting method.

Algorithm 1 Algorithm for RANSAC Depth Plane Fitting

```
1: const  $\theta$ : minimum offset allowed for a single pixel;
2: const  $\epsilon$ : minimum number of inliers allowed;
3: const  $iterations_{max}$ : maximum number of iterations allowed;
4:  $\hat{\beta}$ : Current Model;
5:  $\hat{\beta}^*$ : Optimal Model;
6: for every segment  $C_i$  do
7:    $iterations_{now} = 0$ ;
8:    $error_{best} = \infty$ ;
9:    $\hat{\beta}^* = \emptyset$ ;
10:  while ( $iterations_{now} < iterations_{max}$ ) do
11:    randomly choose three points  $p_1(x_1, y_1, d_1), p_2(x_2, y_2, d_2), p_3(x_3, y_3, d_3)$ ;
12:     $\hat{\beta} = LSE(p_1, p_2, p_3)$ ;
13:     $inliers_{count} = 0$ ;
14:     $inliers_{set} = \emptyset$ ;
15:    for every  $p_i \in c_i$  do
16:      if ( $|[x_i, y_i, 1] \cdot \hat{\beta} - d_i| \leq \theta$ ) then
17:        add  $p_i$  to  $inliers_{set}$ ;
18:         $inliers_{count} ++$ ;
19:      end if
20:    end for
21:    if ( $inliers_{count} \geq |c_i| * \epsilon$ ) then
22:       $\hat{\beta} = LSE(inliers_{set})$ ;
23:       $error_{now} = \frac{\sum_{p_i \in inliers_{set}} |[x_i, y_i, 1] \cdot \hat{\beta} - d_i|}{inliers_{count}}$ ;
24:    end if
25:    if ( $error_{now} < error_{best}$ ) then
26:       $\hat{\beta}^* = \hat{\beta}$ ;
27:       $error_{best} = error_{now}$ ;
28:    end if
29:     $iterations_{now} ++$ ;
30:  end while
31:  return  $\hat{\beta}^*$ 
32: end for
```

4.2.5 Sub-segmentation

A common assumption in segment-based labeling problem is that inside a segment labels should be consistency, however directly allocating disparities in such way

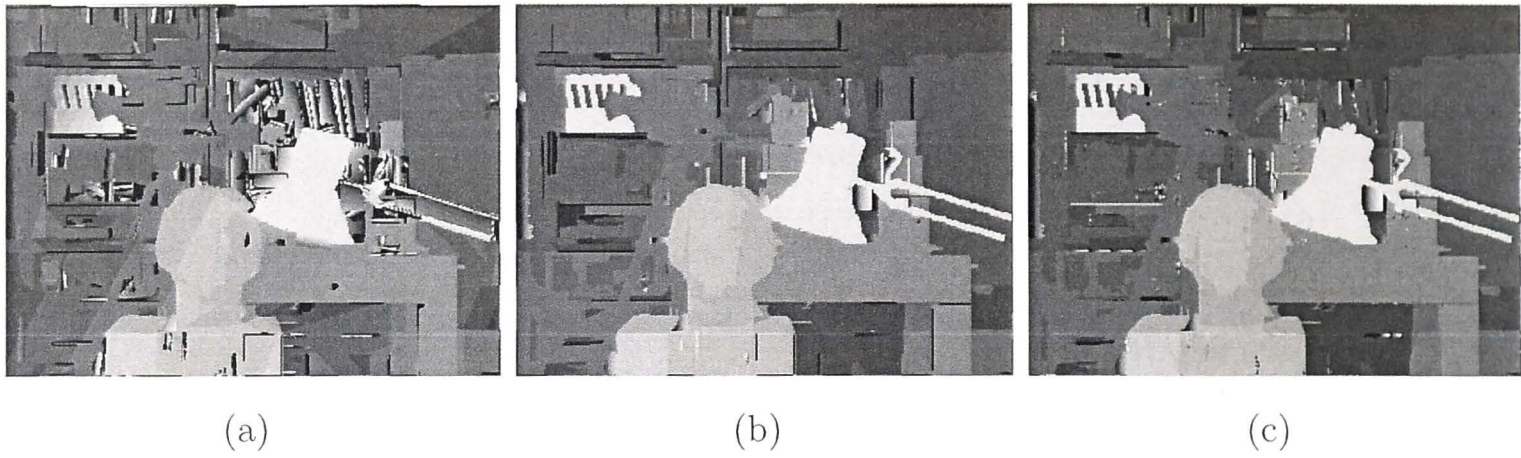


Figure 4.6: Results of three different plane fitting techniques: (a) LSE, (b) RANSAC, (c) Voting-based.

is unreasonable. According to our assumption, we further divide the color-based segments into smaller subsegments so that inside pixels are more likely to share the same depth.

It works as follows. For every plane-fitted segment, we can always extract the planar vector along which the disparity values change the most. Based on the value of each pixel on the surface, pixels can be clustered into sub regions which satisfy the condition that pixels share the same discrete integer value inside the same sub region, as illustrated in Fig 4.7.

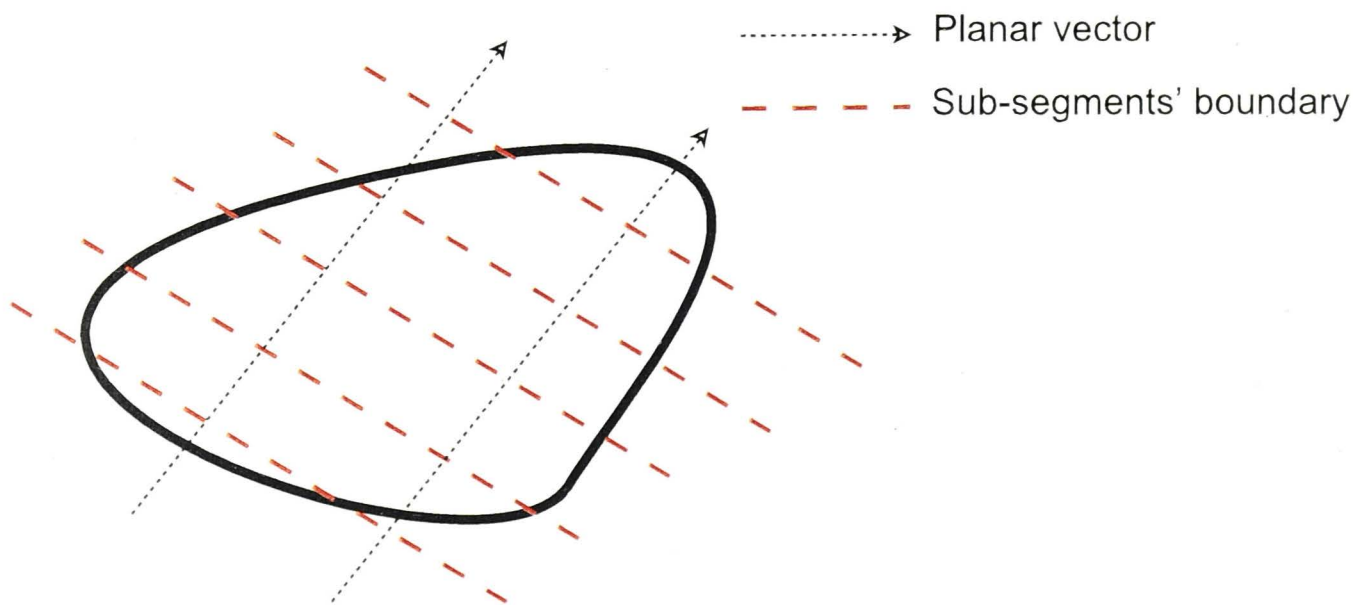


Figure 4.7: An illustration of Sub-segmentation. The black dot lines represent the planar vector, while the red dot lines represent the new sub-segments' boundaries which perpendicular to planar vector. The pixels are separated into sub-segments where pixels inside it have the same discrete value.

4.2.6 Energy Function Model

The framework has been formulated as an energy minimization problem, in which the lowest energy means the globally optimized labeling. In our framework, we add sub-segment information as higher-order potential in the way of a *soft constraint* into our model, and take advantage of the powerful robust higher-order graph cuts algorithm to solve it. The energy function is given by:

$$E = E_{Data} + E_{Smoothness} + E_{HigherOrder} \quad (4.10)$$

The data term is sum over all pixels' local measurement:

$$E_{Data} = \sum_{x_i \in X} c(x_i, y_i). \quad (4.11)$$

The smoothness term is the truncated $L1$ norm function:

$$E_{Smoothness} = \sum_{(x_i, x_j) \in X} \begin{cases} 0, & d(x_i) = d(x_j), \\ \lambda * \min(k, |d(x_i) - d(x_j)|), & otherwise. \end{cases} \quad (4.12)$$

$d(x_i)$ denotes the label of x_i , and k is a truncation parameter. The pairwise form has shown great performance in discontinuity preserving.

In terms of clique(segment/sub-segment) based higher-order term, unlike most existing works formulating higher-order assumption into pairwise terms, we treat all three energy terms equally, which means they are optimized simultaneously. The benefit is that the higher-order term will be able to contain all the assumptions without any sacrifice, and clearly a more globally optimized result will be achieved if three energy terms are treated evenly in the optimization procedure.

We take advantage of the P^n Potts model[35] model, because it meets our assumptions while keeping submodularity. More details will be provided in the next section.

4.2.7 Robust Higher-Order Term and Graph Cuts

According to [24], in order to minimize the energy function by graph cuts, the energy function must be submodular. From additive principle, it is equivalent to that every term of the energy function should be submodular. The data term and pairwise term in our function are submodular, so it is really depend on the higher-order term. And from the definition of submodularity on $F^N (N \geq 3)$, an energy term which involving more than two binary variables e.g., higher-order term, is submodular if and only if all its projections on 2 variables are submodular.

Generally such conditions are hard to satisfy or will need exponential auxiliary nodes added. Thanks to the robust P^n Potts model[35], it is not only submodular but also benefit the inference that only two auxiliary nodes are needed for each clique.

In each sub-segment cli , $Rest(cli)$ represents percentage of pixels not taking the dominant label, then the higher order term is defined as:

$$E_{HigherOrder} = \sum_{cli} \Phi(cli) = \sum_{cli} \begin{cases} Rest(cli) \frac{1}{Q} \gamma_{max}, & \text{if } Rest(cli) \leq Q, \\ \gamma_{max}, & \text{otherwise.} \end{cases} \quad (4.13)$$

In the equation, Q is the truncation parameter which controls the rigidity of this function and satisfies the constraint $2Q < |cli|$, and γ_{max} is the truncated penalty. Here we define γ_{max} as a function inverse to visual feature dissimilarity inside the region:

$$\gamma_{max} = \sum_{x_i \in cli} \frac{(f(x_i) - f(\bar{x}_i))^2}{|cli|}. \quad (4.14)$$

The basic idea is fairly simple, if one segment is less visually homologous, then it is more likely to belong to different depth surfaces, so the penalty for its depth inconsistency should be lower.

It can be seen that the higher-order term is a linear truncated function of the number of inconsistent pixels. While it encourages pixels in one segment to take the same label, it does allow some pixels to take different labels depending on the cost.

According to[35], in α -expansion, the higher-order term can be transformed into sum of first-order and second-order terms:

$$\phi(cli) = \min_{m_0, m_1} ((r_0 \bar{m}_0 + \theta_d m_0 \sum_{x_i \in cli_{dom}} w_i \bar{x}_i + r_1 m_1 + \theta_\alpha m_1 \sum_{i \in cli} w_i x_i - \delta), \quad (4.15)$$

where m_0, m_1 , are two auxiliary nodes, r_0, r_1, w_i , are weight parameters, cli_{dom} denotes the the variables that have been assigned the dominant label in the clique, and δ is a constant.

Hence the higher-order is decomposed into unary and pairwise potential of original variables and two auxiliary nodes, the transformed graph is shown in Figure 4.8.

The details about implementation of higher-order graph cuts is given in Algorithm 2.

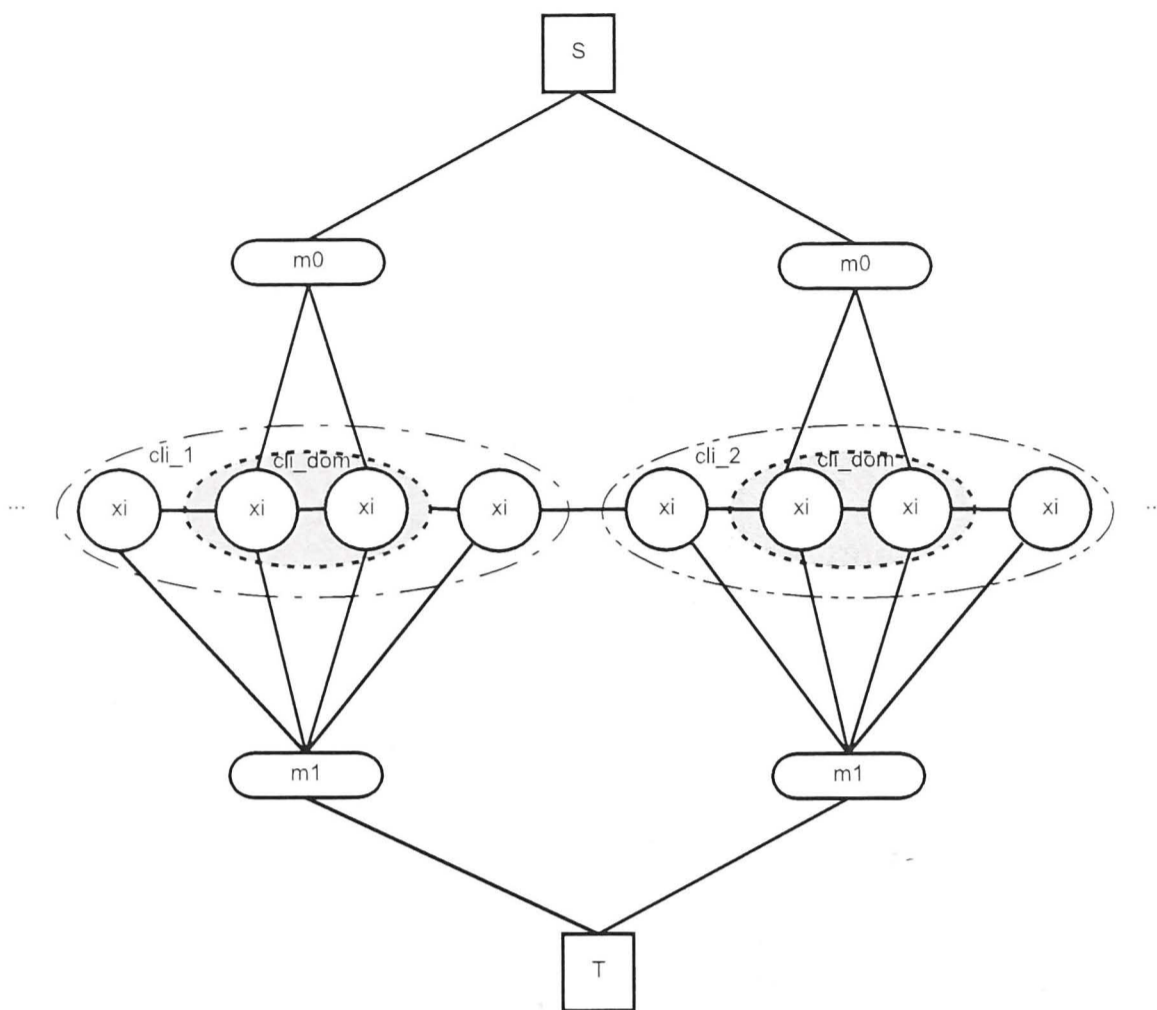


Figure 4.8: Graph construction for higher-order terms. S , T are source and sink. cli_1 & cli_2 here represent two different cliques. Only two auxiliary nodes, namely m_0 & m_1 are needed for each clique.

4.3 Experiment

4.3.1 Quality

To examine the performance of the proposed method, we test on both Middlebury benchmark[67] and challenging real-scene images. The results have been compared with conventional graph cut with the same data term and pairwise term.

From Figure 4.9, the results clearly show that our algorithm succeeds in: (1) It keeps the shape of objects due to sub-segmentation. For example, the arms of the lamp on Tsukuba data set can be clearly distinguished in our result, while it is over-smoothed in conventional graph cut method. (2) It can eliminate ambiguity caused by inaccuracy of initial disparity estimation. This is because the higher-order graph cut process will not only rely on the initial disparity result but also the distribution within segments is taken into account. For example, in the Baby data set, the front of the round object which the baby sits on has some matching

Algorithm 2 Algorithm for Higher-Order Graph Cuts Optimization

```
1: const TimesRemain =  $\tau$ ;
2: LabelNow = {0};
3: while TimesRemain > 0 do
4:   for random  $i \in L$  do
5:      $\alpha = i$ 
6:      $E = \text{ComputeEnergy}(\text{LabelNow})$ 
7:     for every  $x_i \in X$  that  $\text{LabelNow}[x_i] \neq \alpha$  do
8:       add node  $x_i$  and its t-links
9:     end for
10:    for every pair  $\{x_i, x_j\} \in X$  that  $\text{LabelNow}[x_i] \neq \alpha \ \&\& \ \text{LabelNow}[x_j] \neq \alpha$  do
11:      add n-link between  $x_i$  and  $x_j$ , also update t-links
12:    end for
13:    for every  $cli \in X$  do
14:      add auxiliary nodes  $m_0, m_1$  and their t-links
15:      for every  $x_i \in cli$  that  $\text{LabelNow}[x_i] \neq \alpha$  do
16:        add n-link between  $x_i$  and  $m_0$ 
17:      end for
18:      for every  $x_i \in cli$  that  $\text{LabelNow}[x_i] = \text{DominantLabel}$  do
19:        add n-link between  $x_i$  and  $m_1$ 
20:      end for
21:    end for
22:    Apply max-flow algorithm and update LabelNow
23:     $E' = \text{ComputeEnergy}(\text{LabelNow})$ 
24:    if ( $|E' - E| < \text{eps}$ ) then
25:      TimesRemain --
26:    else
27:      TimesRemain =  $\tau$ 
28:    end if
29:  end for
30: end while
31: return LabelNow
```

errors by traditional graph cut method, but is recovered as part of a plane in our algorithm. (3) High accuracy is achieved in typical difficult areas such as textureless regions. An example can be found in the real-scene image where the

Algorithms	Tsukuba			Venus			Teddy		
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
Regular GC	2.27	4.36	11.4	1.27	2.90	13.4	14.9	23.6	24.7
Proposed method	2.01	4.01	9.75	0.74	1.88	8.54	11.4	18.0	23.5

Table 4.1: Error rate on Middlebury Benchmark. It can be easily seen that our proposal outperforms regular graph cut in all three indicators.

left-down part of the ground is textureless, our algorithm handles it accurately. Table 4.1 of error rates quantitatively describes the performance of our method in comparison with the regular graph cut. It clearly demonstrates our approach outperforms in all three indicators with average improvement of more than 20 percent.

4.3.2 Efficiency and Energy Convergence Analysis

The increased auxiliary nodes only take a small proportion of the original nodes, therefore there is almost no extra time consumption for each max-flow iteration. So the efficiency here is all related to the number of iterations taken to converge, here one iteration is referred to α visits every label in L once.

From Figure 4.10, it can be observed that most of the energy are minimized within the first iteration, and all three Middlebury sample images converge well within 5 iterations.

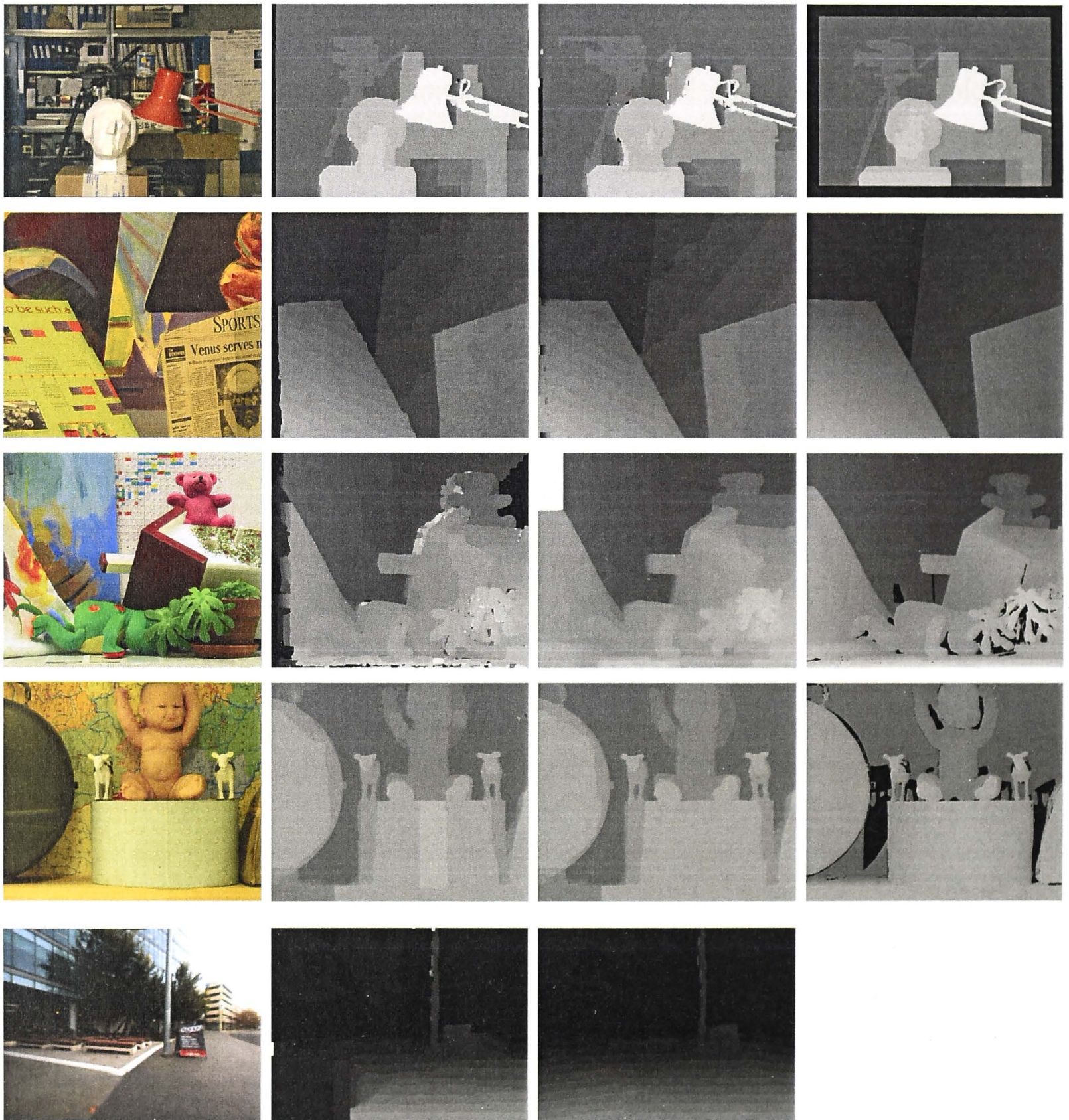


Figure 4.9: Results on Middlebury(Tsukuba, Venus, Teddy, Baby)& Real-scene data. From left to right: image input, regular graph cut, our result, ground truth.

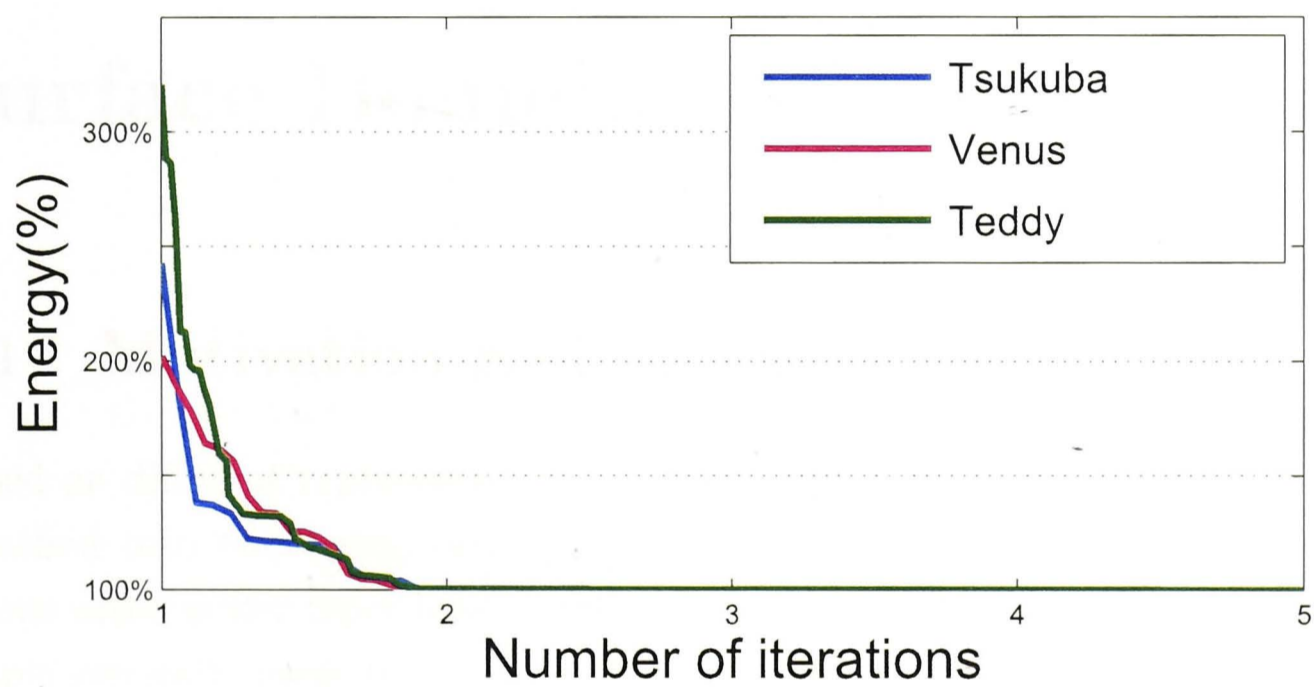


Figure 4.10: Energy minimization on three Middlebury images. We take the final converged energy as 100%. And one iteration involves α visiting every label in label set once.

Chapter 5

Joint Optimization on Coupled MRFs for Stereo Matching and Surface Boundary Estimation

5.1 Motivation and Introduction

Based on different representations of depth estimation, existing methods can be classified into two categories: pixel-wise and segment-wise. Pixel-based algorithms often suffer from local noises and have insufficient cues of the scene. As people generally identify the object and reconstruct the scene by partitioning the scene into a set of groups each with the same or similar visual features such as color or texture, researchers have developed segment-based algorithms based on the similarity.

Segment-based algorithms [73][78] have dominated the Middlebury Benchmark [67] due to their good performance on reducing ambiguity of disparities in textureless regions. They usually share the assumption that the scene structure can be approximated by a set of non-overlapping visually homogeneous regions where each region corresponds to its own depth surface. In other words, all pixels in the same segment should lie on the same depth surface and discontinuities only occur on boundaries. This assumption certainly enhances the tolerance of local noise as the depth surface is now decided by a group of pixels, the risk of assigning incorrect disparities to occluded or textureless individual pixels is decreased. Typical procedures for these approaches are as follows: first, segmenting the reference image using color-based segmentation and getting an initial disparity by doing pixel-based local match; then fitting disparity planes to every

segment using plane fitting techniques; finally the optimal assignment of planes is approximated by using global-based optimization tools to minimize a certain energy function.

However, this assumption has some drawbacks. First, with segments being purely grouped on visual features, they are still likely to be influenced by local noises. Imagining a piece of colorful newspaper lying on a planar table. Clearly the newspaper should locate on the same planar depth surface. However in segment-based algorithms, every individual character and color region may be segmented into different sized segments. Segment-based approaches usually are not concerned with the dimension of the segment, and simplify each segment as an individual node in the model for further optimization. Therefore robustness will not be guaranteed due to the existence of these small segments. Second, Color segmentation only relies on visual cues, but the correspondence between visual features and depth does not always hold. Twoneighboring surfaces with huge difference in depth but little variance in color sometimes are segmented as one segment, which resulting in assigning one faulty surface for both. A typical example is shown in Fig 5.1. Third, although the first phenomenon may be regularized by adding smoothness interaction between neighboring segments, this is under the assumption that the depth is spatially smooth everywhere, even for the neighboring segments that actually cross the surface boundaries. As a result, the parameter of smooth scale is always hard to tune. If the parameter is too small, these small segments will not be as consistent as desired, but if too large it will lead to undesired blurring along surface boundaries because the neighboring segments that actually cross the surface boundaries are smoothed as well. An alternative solution is to introduce depth surface boundaries to distinguish the smoothness of neighboring segments along the surface boundaries. An experiment motivates us is that given perfect or near perfect surface boundaries, state-of-the-art results can be achieved by over-smoothing segments within the same depth surface.

In addition, low resolution is crucial for some specific applications for artificial visual simulation[55][54][75]. Under the present hardware limitation of low-vision devices, the depth must be down-sampled to a qualified low-resolution. Apparently, some popular image resizing methods(nearest-neighbor, bilinear, cubic and so on) will be the straight-forward solution, but they may bring some serious distortions into the results in which the surface boundaries are blurred, and depth of foreground merges into background. This is partially due to the equally treatment of boundary regions and no boundary regions. Down-sampling within the

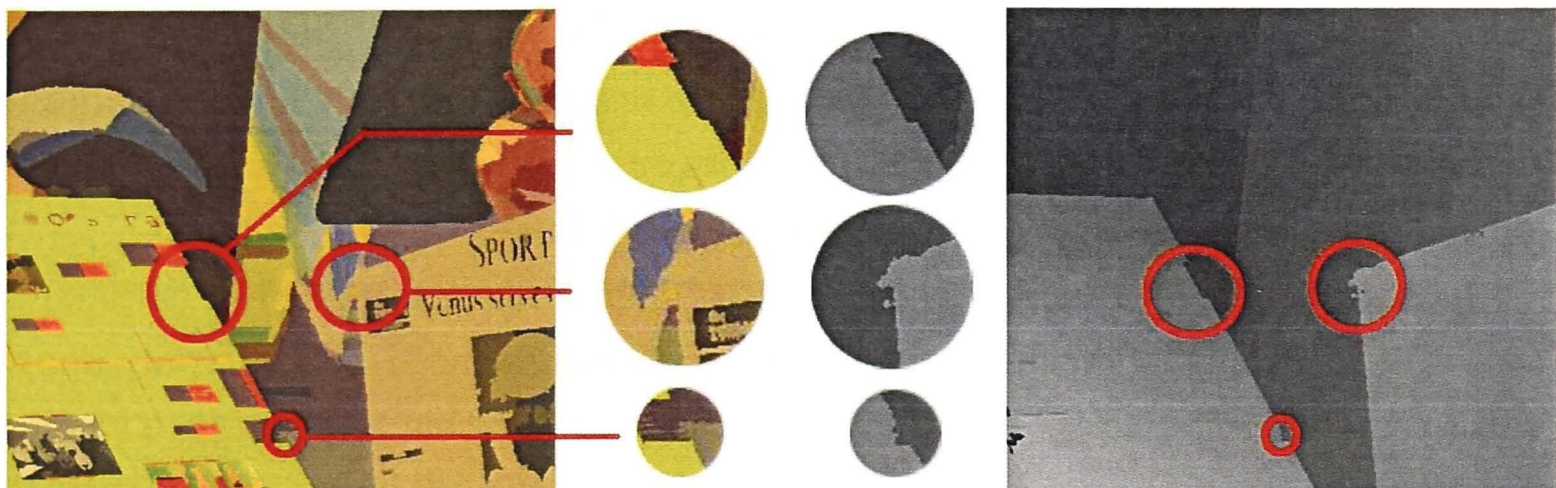


Figure 5.1: An example of the Venus image pair from the Middlebury Benchmark[67]. The image on the left is the color segmentation by Mean-Shift[16], the image on the right is one typical state of the art result[85]. From the correspondence, it can be clearly concluded that three major errors are all caused by inaccurate initial color segmentation.

same depth surface is straightforward and easy to implement, but boundary regions should be handled carefully. Therefore, surface boundary can be used as clues into down-sampling process.

The above challenges motivate us to integrating depth surface boundary estimation into the existing stereo matching framework so that these two types of variables could be inferred together and interact each other. In the chapter, we will present two novel approaches that employs a similar two-layer MRFs framework. The first is geometry-based, and has shown great performance in surface boundary completion. The second is natural boundary-based and we have successfully applied it in phosphene vision for indoor human navigation.

Inspired by Ren's work[59], we use one layer to model the connectivity of locally found edges. In his work, he builded a *constrained Delaunay triangulation*(CDT) over the locally found edges and used Conditional Random Field to model the continuity of edge junctions. Because the connectivity of the CDT edges involved higher-order clique, he used loopy belief propagation[74](LBP) to estimate the marginal distributions. But in our framework, with a two layers MRFs the graph will be much more complex so that LBP will be computational expensive. As a result, the connection between boundary nodes are simplified from higher-order to pairwise relationship. After these two layers are modeled separately, we align and associate two layers based on the topological structure.

Our first approach also take advantage of the CDT, and we will demonstrate that this geometry-based modeling has significant advantage in surface boundary

completion. Besides that, we also propose another approach that novelly break boundaries into pieces so that two neighboring segments will only have one unique boundary piece between them. And we treat such boundary pieces as individual variables in the boundary layer of associative MRFs.

In both approaches, along with surface boundaries determined dynamically, smoothness scaling between segments can be decided as need, and will only apply within surface boundaries. In some sense, it can be seen that segments are formed dynamically according to boundaries. And both surface boundary and depth obtained simultaneously facilitates further recognition and scene understanding.

Generally, optimizing such framework is quite complex and challenging. The third-order interaction between two layers makes standard graph cuts approach difficult to apply. Also dense short loops will lengthen the time taken to converge in message-passing algorithms. Thanks to the latest projected graph cuts[39], it minimize the energy by making projected moves iteratively, in which it fixes one layer of MRFs at a time, and uses ST-min cut[11] to optimize the other layer. It converges when no lower energy can be reached.

Experiments demonstrates our novel approaches could provide 1) significant improvements by eliminating depth ambiguities and increasing its accuracy, 2) explicit clues of depth and boundary for human navigation under low-resolution phosphene vision, 3) foreground obstacles are clearly discriminated from surrounding background by integrating boundary clues into downsampling process.

5.2 Triangulation-Based Joint Framework for Stereo and Surface Boundary Completion

In our first approach, we build a two layers MRFs that modeling depth and its surface boundary simultaneously. In the experiment, we find that the geometry nature of *constrained Delaunay triangulation*(CDT) makes it capable of completing true surface boundaries which missed by local edge detector. As well as due to efficiency, our first approach is carried out on triangle-based instead of pixel-based. Sides of triangles are defined as variables for surface boundary with only two states, on and off. And the set of pixels that each triangle covers are defined as one variable for stereo matching. An illustration is in Figure 5.2.

More formally, two sets of variables are used, X for depth and Y for boundary. Let $L_x = \{1, 2, \dots, n\}$ be a set of n different discrete depth plane labels, and $L_y = \{0, 1\}$ be a two-variable set for the labels of surface boundary in which 0

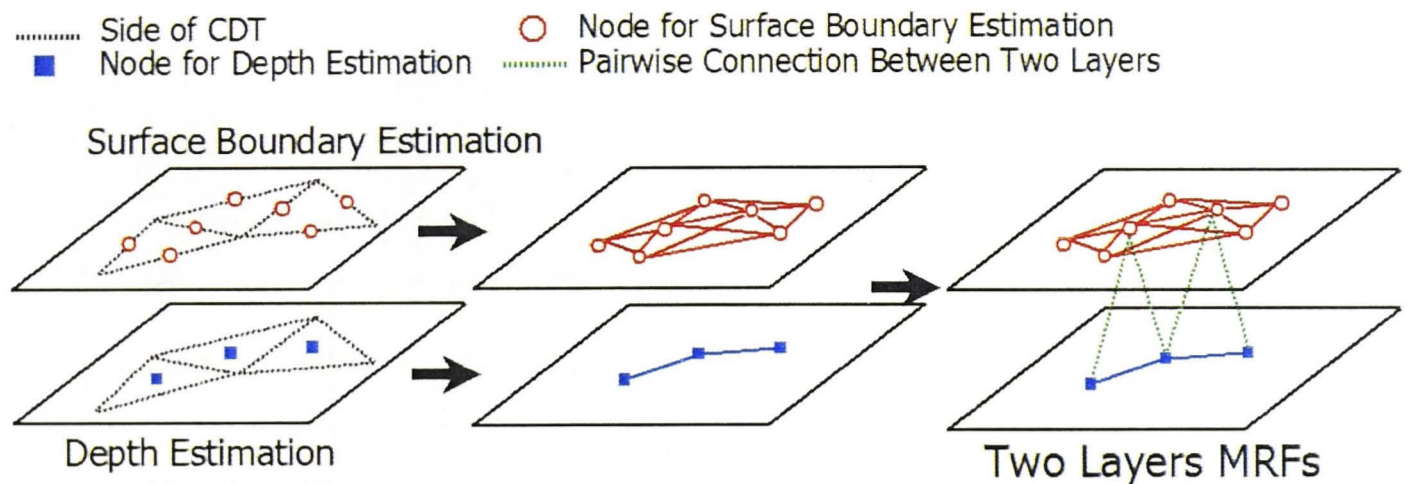


Figure 5.2: Triangulation-based two layers MRFs framework.

is off and 1 is on. The task is to find a labeling configuration f that allocates the labels from L_x to each variables $X_i \in X$ and L_y to each $Y_i \in Y$ respectively. Then each possible labeling f has its own *posterior* probability, the goal is to find the f^* that has the maximum probability. According to the Hammersley-Clifford theorem, maximum a *posterior* labeling f^* (MAP) is equivalent to the minimum of the Gibbs energy. We define the proposed energy function as:

$$E = \underbrace{E^S(x)}_{\text{Stereo}} + \underbrace{E^B(y)}_{\text{Boundary}} + \underbrace{E^I(x, y)}_{\text{Interaction}}. \quad (5.1)$$

This energy function not only contains energy potentials $E^S(x)$ and $E^B(y)$ for stereo matching and boundary estimation alone but also has energy term regarding the interaction between them. Details will be given later.

The main steps of our algorithms are illustrated in Figure 5.3.

5.2.1 Boundary Potentials

Our approach starts with locally captured edges by probability of boundary detector[47], then the probability for every edge is normalized into $[0, 1]$. After that, we break up the boundaries into piecewise linear segments at high-curvature locations. To do this, we trace each boundary from one conjunction point to the other, and recursively split the curve into approximate line segments to satisfy that the angle between two splits will always exceed a certain threshold. Once the decomposition has been completed, we have a set of conjunction nodes and a set of line segments. Each line segment will be given an probability value pb that equals to the average probability of all the pixels its corresponding curve passing through.

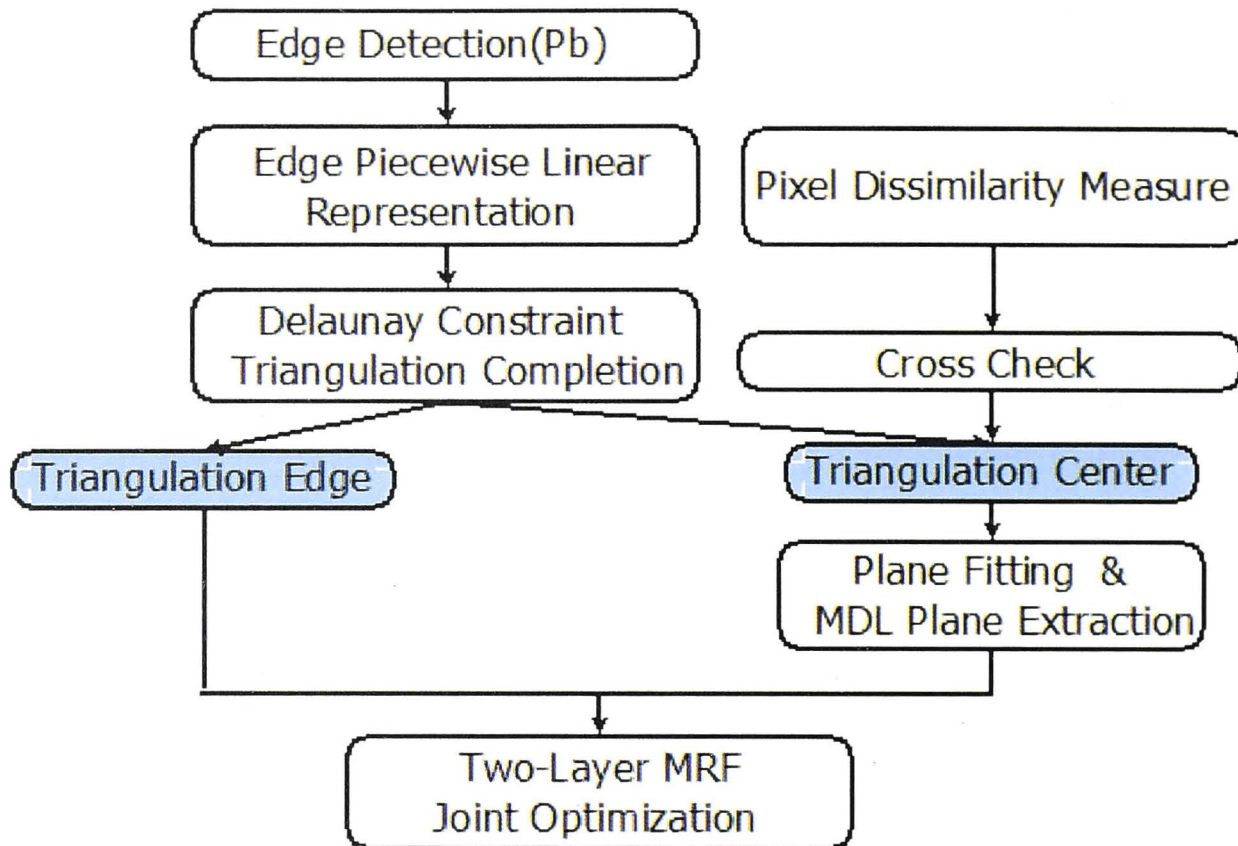


Figure 5.3: The flowchart of our triangulation-based algorithm.

Next we employ the *constrained Delaunay triangulation*(CDT) algorithm to predict the missing edges. CDT is a generalization of the standard Delaunay Triangulation that forces the generated triangulations passing through certain required segments(in our case, the edges in the piecewise linear approximation). Here the pb value for these completed edges are given as 0. An example of these processes is given in Fig 5.4.

5.2.2 Surface Boundary Potentials

The energy potentials for surface boundary estimation is defined as

$$E^B(y) = \psi_i^B(y_i) + \psi_{ij}^B(y_i, y_j) \quad (5.2)$$

The unary term $\psi_i^B(y_i)$ only penalize on the situation that the boundary choose to be appear. The lower its local probability is the higher penalty it will take:

$$\psi_i^B(y_i) = \sum_{y_i \in Y} (1 - pb_i) \cdot y_i. \quad (5.3)$$

The pairwise term $\psi_{ij}^B(y_i, y_j)$ encourages two connecting boundaries to be both turned on or turned off. Define Θ_{ij} as the angle between two edges, when $\Theta_{ij} \rightarrow \pi$, it indicates a strong continuity and the possibility for them to have the

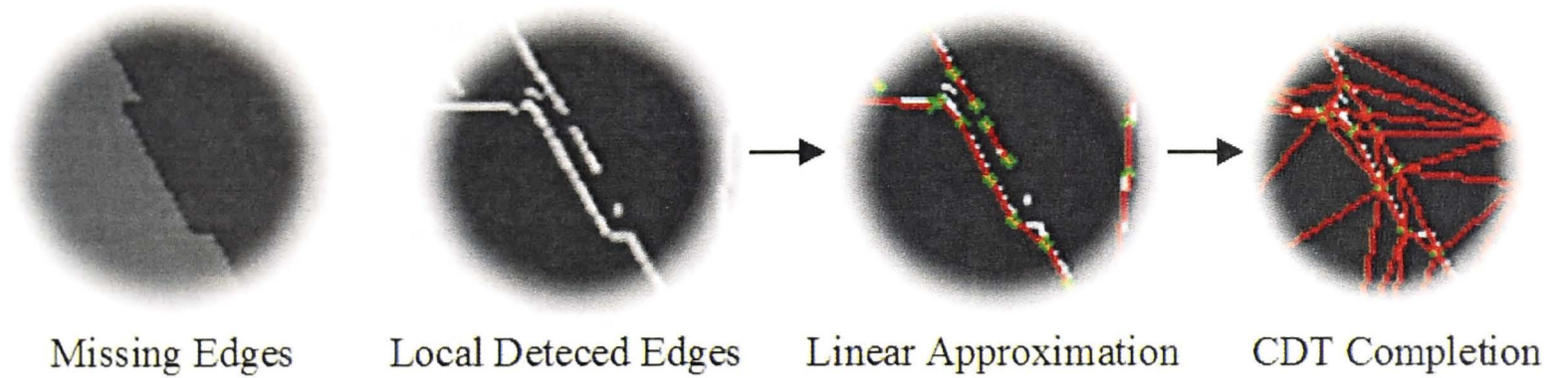


Figure 5.4: An example demonstrates the process and advantage of edge completion. From left to Right, the first image is the depth result with missing edge (inaccurate segmentation). The second image is the result of local edge detection. In the third image, the red edges represent linear approximation, the green dots represent line segment terminals. The fourth image is the CDT completion result. It can be seen that it successfully complete the missing edge.

same states should be greater, vice versa. It takes the form of a data-driven Potts model:

$$\psi_{ij}^B(y_i, y_j) = \sum_{y_i, y_j \in N} \begin{cases} 0, & \text{if } y_i = y_j, \\ \Theta_{ij}, & \text{otherwise.} \end{cases} \quad (5.4)$$

5.2.3 Stereo Matching Potentials

To begin with, we use the fast local pixel dissimilarity measure [7] to construct the correlation volume for both left and right images as the reference image. Then we apply mutual consistency check on the result. Pixels passing it will be labeled as *stable pixels*. The reasons for failing mutual consistency check include occlusion, textureless and specific faulty matching.

Once obtained initial depth and a set of *stable pixels*, a RANSAC plane fitting is carried out inside each triangle with the depth of the left view. Note we only apply RANSAC on *stable pixels*, and also only choose to implement in a triangle if its percentage of *stable* members exceeds a certain threshold. For every implementable triangle, we put the computed plane with the least error into L_x , and keep records of its frequency f_{l_x} .

Plane Extraction with MDL Regularization

As the fitting label L_x having too many members, it not only slows down the final global optimization but also arises more noise. To cut down the volume, we add a plane extraction step to merge neighboring planes.

The energy function for plane extraction is defined as:

$$E_{MDL} = \psi_i(x_i) + \psi_{ij}(x_i, x_j) + \underbrace{\sum_{l \in L_x} \gamma_l \cdot \delta_l}_{\text{label cost}}, \quad (5.5)$$

where $\psi_i(x_i)$ is the sum of pixel-based absolute depth difference between its original plane and new mapping plane. $\psi_{ij}(x_i, x_j)$ is a Potts model penalizing on difference. The label cost term [17] functions as a MDL regularizer, penalizing on occurrence of a certain plane, and is a decreasing function of the frequency f_{l_x} in the RANSAC result. More formally, we use $\gamma_l = e^{-f_{l_x}}$, and

$$\delta_l = \begin{cases} 1, & \exists x, l_x \in L, \\ 0, & \text{otherwise.} \end{cases} \quad (5.6)$$

Roughly, the size of L_x is cut down to less than 20 after this step.

The energy function for stereo is then defined as:

$$E^S(x) = \psi_i^S(x_i) \quad (5.7)$$

The unary term $\psi_i^S(x_i)$ is the sum of absolute difference between current labeling and initial disparity map. We do not have a conventional pairwise term for stereo here is that we modified it into an interaction term with boundary, it will be described in details in next section.

5.2.4 Interaction Potentials

For each pair of neighboring x_i and x_j there will be an unique piece of boundary namely y_k . The interaction potential is defined as:

$$E^I(x, y) = \sum_{x_i, x_j, y_k} \psi_{ij}(x_i, x_j) \cdot \bar{y}_k \quad (5.8)$$

where $\psi_{ij}(x_i, x_j)$ is a Potts model. The principle of the projected graph cuts is to fix one layer in MRFs at a time while optimizing the other. When layer X is fixed, and neighboring x_i and x_j do not belong to the same depth surface ($\psi_{ij}(x_i, x_j) = 1$), the energy potential will intend to decrease itself by encouraging the boundary between to be appeared ($y_k = 1$). And when layer Y is fixed and y_k is turned on, the energy potential will be 0 thus the smoothness requirement of x_i and x_j will no longer be executed.

5.2.5 Joint Inference

This two-layer MRFs have the set of variables up to $\{X, Y\}$ and label space up to $L_x * L_y$. Graph with such complexity is generally difficult to optimize. We bring the idea of Projected graph cuts (PGC) [39] to α -expansion optimization, it gives an approximation of the true labeling at an acceptable efficiency.

The requirement for using PGC is that the potential defined between two layers (in our energy function, the interaction term $E^I(x, y)$) should always be projected submodular, that is when fixing one layer of variables, the other layer should satisfies submodularity. In our case, fixing X will make the rest of the energy term as a first order term for Y , so it is always submodular. On the other hand, when fixing Y , the rest term will either be 0 or Potts model with positive weights, so it is submodular too.

The basic steps for the inference is as follows. We start randomly either from the initial labeling f_X or f_Y , and do the optimization recursively. For instance, when we optimize Y in one iteration, suppose the optimal labeling achieved so far are f_X^* and f_Y^* . We fix X in $E^I(x, y)$ by taking the values from f_X^* , and put the transformed term together with the stand alone term $E^B(y)$, and use ST-min cut to optimize variable Y alone. If a lower energy with solution f_Y' is found, we keep the f_X^* unchanged and set $f_Y^* = f_Y'$. Optimizing X is applied in a similar subsequent way. When no lower energy can be achieved in $L_x * L_y$ iterations, the optimization stops and returns f_X^* . Details are given in Alg. 5.2.5.

5.3 Segment-Based Joint Framework for Phosphene Vision in Indoor Navigation

For indoor navigation purpose, we have employed the framework to present another approach on segment-level. Comparing to the first approach, it has more accurate boundaries and higher efficiency. In addition, when integrating the boundary clues into downsampling process, the foreground obstacle has been clearly enhanced and discriminated from the surrounding background.

The framework is in general similar. The first stage of the proposed approach is color segmentation[16] on the reference image. For stereo matching, every segment is taken as an individual depth node disregard of their sizes. And for each pair of neighboring segments, define their unique piece of boundary connection as one boundary node. An illustration of this process is given in Figure 5.5.

Algorithm 3 Algorithm for Projection Graph Cuts Optimization in Our Framework.

```

1: const  $TimesRemain = \tau$ ;
2:  $f_x = \{0\}$  and  $f_y = \{0\}$ ;
3:  $E = ComputeEnergy(f_x, f_y)$ 
4: while  $TimesRemain > 0$  do
5:   for random  $L_x$  do
6:     Fix  $f_y$ , transform  $E^I(x, y)$  to  $E^I(x, f_y)$ 
7:     Add  $E^I(x, f_y)$  to  $E^S(x)$ 
8:     Apply  $\alpha$ -expansion and get the newest labeling  $f'_x$ 
9:      $E' = ComputeEnergy(f'_x, f_y)$ 
10:    if  $E' < E$  then
11:       $E = E'$ 
12:       $f_x = f'_x$ 
13:       $TimesRemain --$ 
14:    else
15:       $TimesRemain = \tau$ 
16:    end if
17:  end for
18:  for random  $L_y$  do
19:    Fix  $f_x$ , transform  $E^I(x, y)$  to  $E^I(f_x, y)$ 
20:    Add  $E^I(f_x, y)$  to  $E^B(y)$ 
21:    Apply  $\alpha$ -expansion and get the newest labeling  $f'_y$ 
22:     $E' = ComputeEnergy(f_x, f'_y)$ 
23:    if  $E' < E$  then
24:       $E = E'$ 
25:       $f_y = f'_y$ 
26:       $TimesRemain --$ 
27:    else
28:       $TimesRemain = \tau$ 
29:    end if
30:  end for
31: end while
32: Set  $f_x^* = f_x$  and  $f_y^* = f_y$ 

```

The definition of the energy function is similar to our first approach, and we also use PGC for optimization. After obtaining the depth and surface boundary

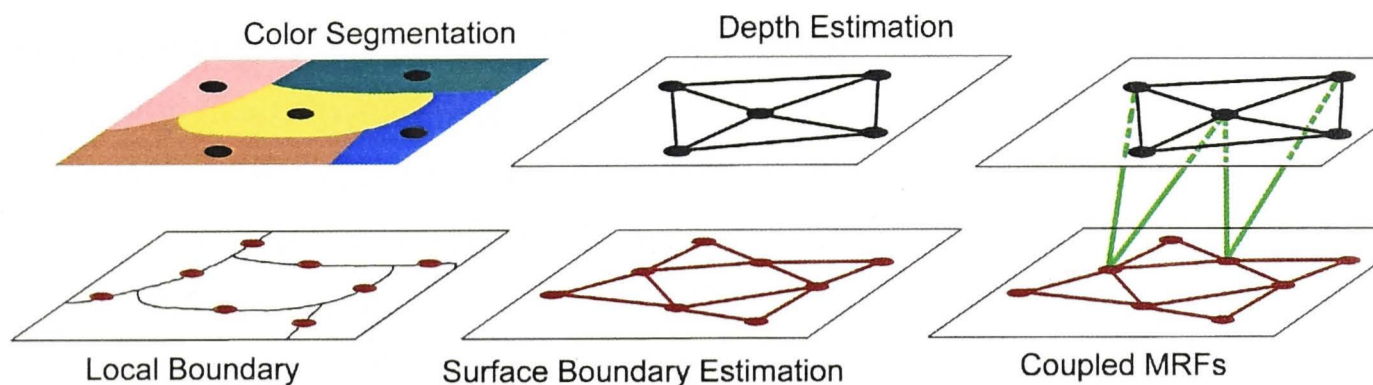


Figure 5.5: The proposed two-Layer MRFs Framework. We use color segmentation as our inputs. For depth estimation in the upper-layer, every segment is modeled as one node(black). For surface boundary estimation in the downer-layer, boundaries are further broken into piecewise ones(red). The green lines are the connection between two layers. For simplicity, here only draws the two-layer connections(green) of two boundary nodes.

result, the downsampling and phosphene visualization process are carried out to convert the depth into phosphene vision in order suit human navigation.

5.3.1 Downsampling and Phosphene Representation

There exists a variety of image down-sampling methods. Interpolation of bilinear and cubic will compose new values for anti-aliasing purpose which may cause confusion in depth-based human navigation. Although simple nearest neighbor will not add new value, it is not robust for low-vision navigation either as it may omit some critical information in the foreground. Here we propose a novel down-sampling method by integrating the boundary clues to the down-sampling process, which clearly help to discriminate the obstacle object from the surroundings in phosphene-based low-resolution navigation trial.

A brief example is given in Figure 5.6. The principle of nearest neighbor down-sampling is to project every down-sampled node(pixel) to original image and obtain its sub-pixel location and coordinates, and then simply select the value of its nearest neighbor as its own. However in low-vision navigation, the priority is to avoid the nearest obstacles. Therefore during the down-sampling process, nearest neighbor algorithm may omit some critical information of foreground obstacles which merged into background, and this will cause serious problems in navigation. Such errors always happens in surface boundaries where the depth significantly changed. To solve it, we have modified and improved nearest neighbor algorithm by integrating the boundary clues to efficiently solve the

problem. During the down-sampling process, it takes advantages of the boundary map, for the sub-pixel projected in the original image, if any of its neighbors in a limited scope locating on the boundary, the sub-pixel will take the largest depth value among its neighbors, otherwise it takes the value of its nearest neighbor. The experiments demonstrates such modification could emphasize the foreground objects significantly in low-resolution vision.

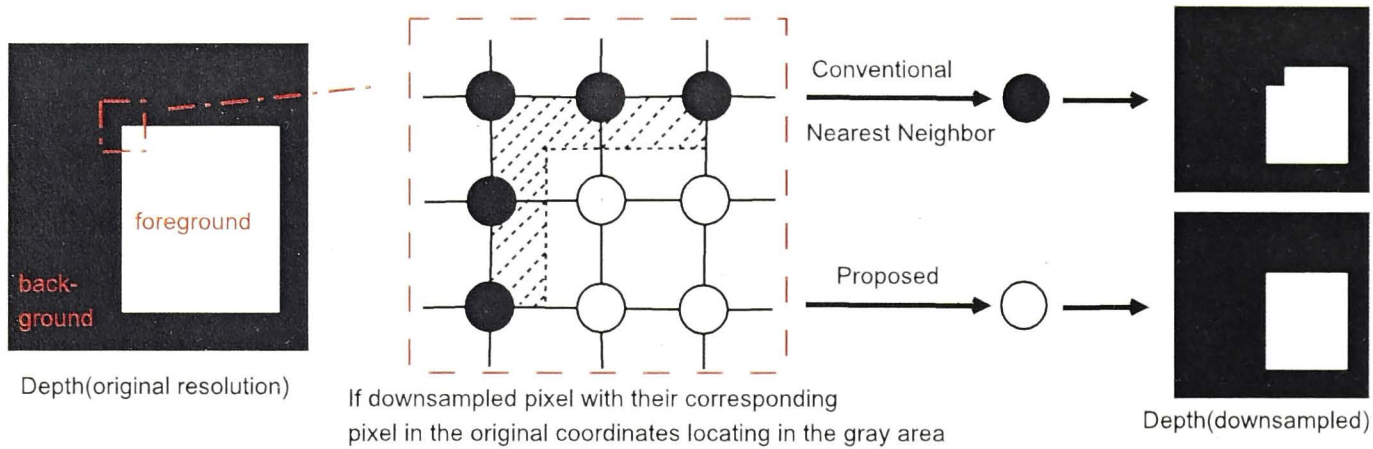


Figure 5.6: An example demonstrates the advantage of our downsampling algorithm comparing to the conventional nearest neighbor.

For stimulated phosphene rendering after down-sampling, each phosphene is represented by a circular Gaussian whose center value and standard deviation are modulated by the depth at that point. In addition, phosphene sums their values when they overlap. For complete description, please refer to [45].

5.4 Experiment

Two proposed methods have been tested on Middlebury's benchmark images[67] and our indoor navigation real-scene dataset. The testbed is on a desktop computer with Intel core I3 2.93Ghz CPU. In the first approach, the CDT function is realized by calling the Matlab function in Microsoft Visual Studio, and most of the time is spent on this procedure. For the PGC optimization, it takes less than 60 seconds to process a high-resolution image pairs. The second algorithm is with higher efficiency, the time has shorten to 100 seconds, and this includes the time taken by all the processes.

5.4.1 Experiment of Triangulation-Based Algorithm in Surface Completion

Here we provide the results on two representative image pairs, Venus and Map[67] in Fig 5.7. From the results, it can be observed that our algorithm succeeds in

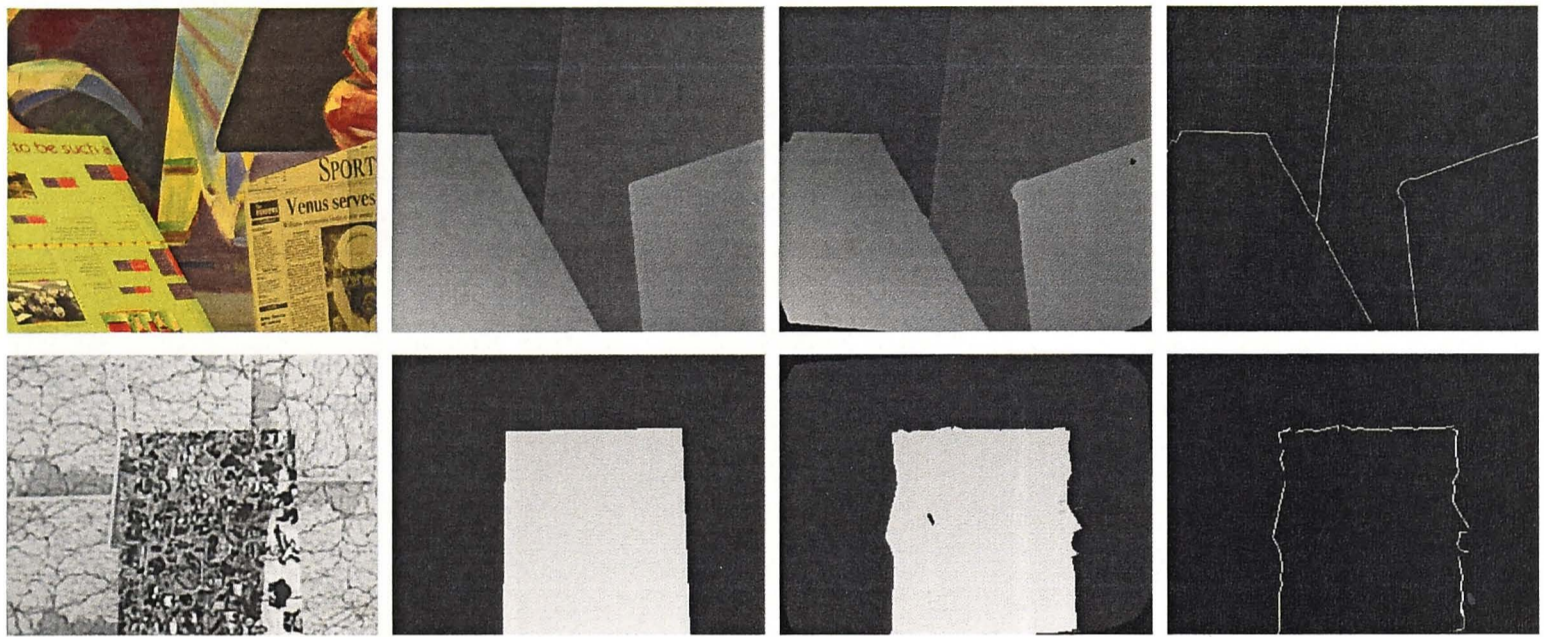


Figure 5.7: From left to right : left image from image pair, depth ground truth, our depth result, our surface boundary result

capturing surface boundaries. In Venus, it clearly distinguishes the scene into four individual depth surfaces. With such information, it provides convenience for further high-level vision works. Due to the noise by local stereo correlation, the boundaries and depth are not perfect in our case, however its accuracy is still comparable to state of the art results. Quantitative measurement is given in Table 5.1.

nonocc	all	disc
0.23	0.43	2.77

Table 5.1: Our accuracy on Venus.

5.4.2 Experiment of Segment-Based Algorithm in Human Navigation

For experiment of the second algorithm, the analysis on the real-scene dataset is presented in Figure 5.8 and Figure 5.9, while the comparisons on the Middlebury's images are in Figure 5.10 and Figure 5.11.

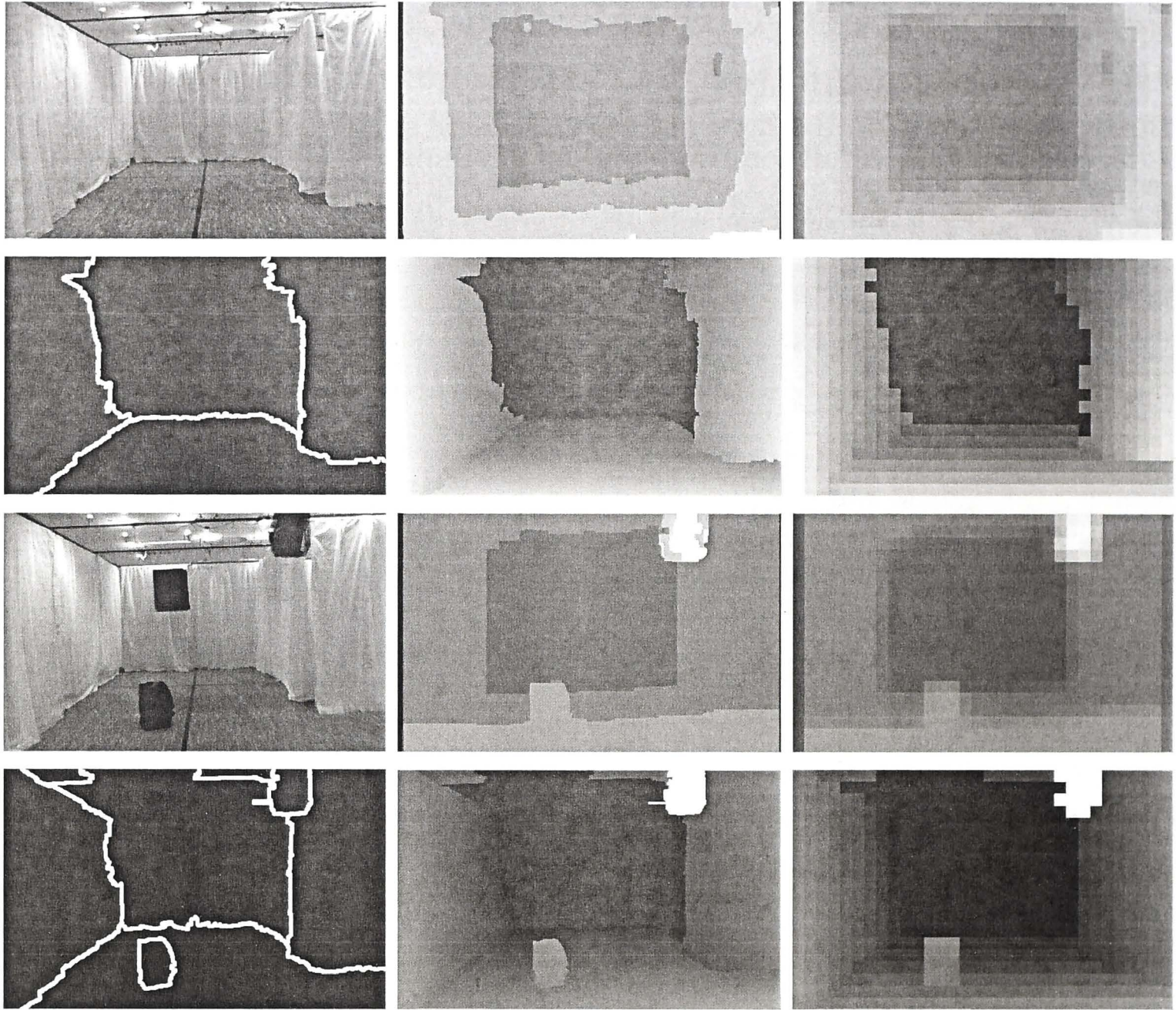


Figure 5.8: The first row includes the original image without obstacles, its original-size and downsampled depth computed by Graph Cut, followed by second row with the results obtained by our algorithm, respectively surface boundary, depth and its downsamples. The third and fourth rows are the results of the images with obstacles. (Original image size: $500 * 312$, downsampled image size: $32 * 20$)

From the results of the indoor image pairs in Figure 5.8, It clearly presents that our approach has more natural and continuous depth than traditional graph cuts under both obstacle and non-obstacle image pairs, as well as the obstacles stand discriminatively from the background. When comparing the performance of downsampled results, the obstacle objects are clearly discriminated from the surroundings after integrating the boundary clues into down-sampling process and it is valuable for further object detection use. While the obstacles in the traditional down-sampled look vague. In Figure 5.9 of zooming out interest regions, those obstacles could be more clearly observed in phosphene visualization.

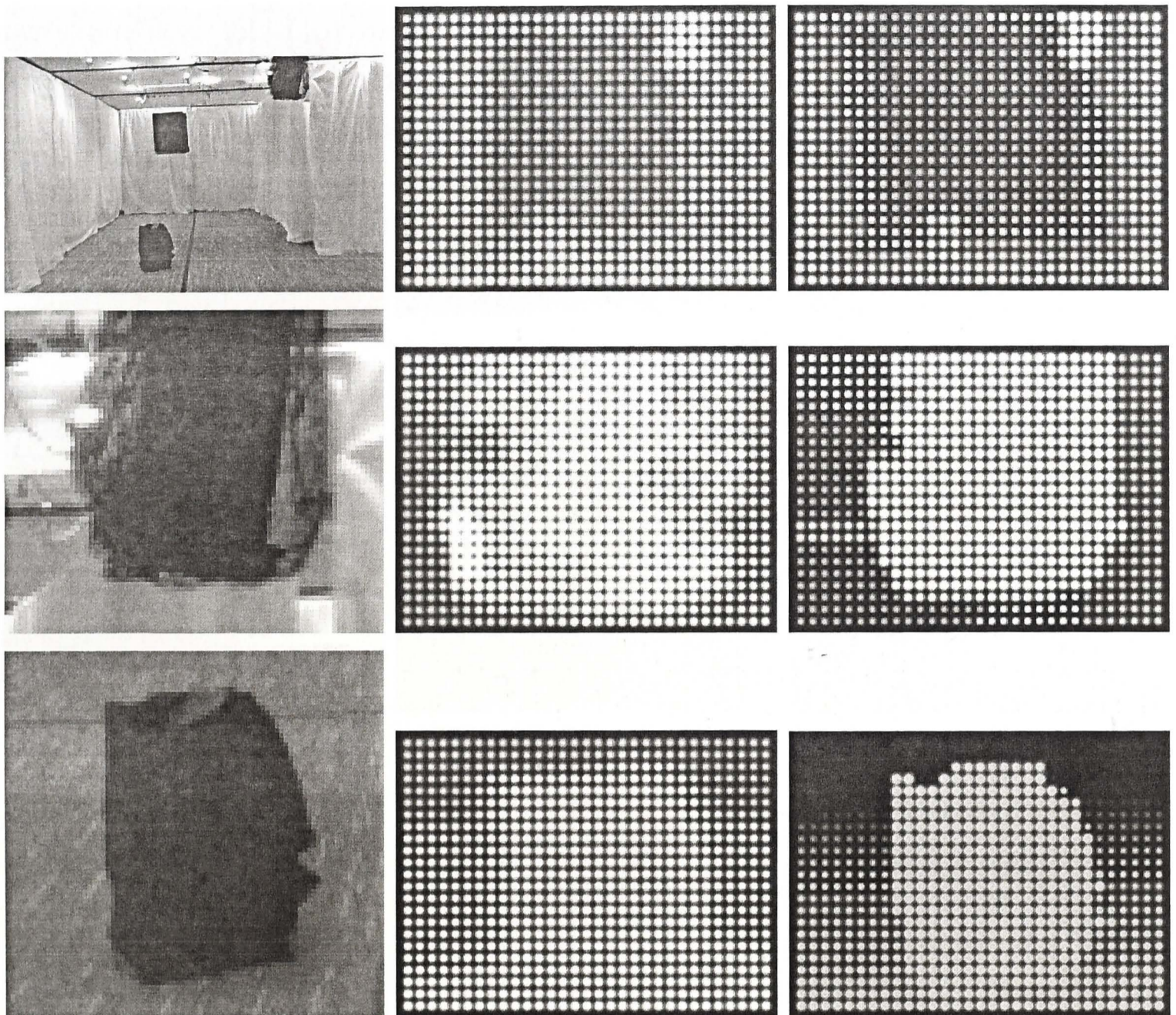


Figure 5.9: Phosphene simulation of indoor scene with obstacles. The first row uses the full camera size image as the input, while the last two rows are the obstacles zoom-in effect which could be crucial in real navigation. The second and third columns are the result by Graph Cut and the proposed algorithm respectively. It can be seen that the latter one has obvious advantage in obstacle distinction.

For quantitative analysis, the proposed method has been tested together with conventional graph cuts and belief propagation methods on four classic Middlebury image pairs Venus, Teddy, Cones and Tsukuba, under three different scales of original size, 1000 and 100 samples respectively. The accuracy is calculated in the following way. For every unoccluded pixels, the absolute difference of their depth with ground truth is calculated. Pixel with difference large than 1.0 will be labeled as *bad pixel*. The error rate is the average percentage of these *bad pixels* over all unoccluded pixels in two Middlebury images. The original ground truth and occlusion map are all down-sampled to align the comparison under

difference scaling. The results of Figure 5.10 and Figure 5.11 clearly demonstrate our method outperforms other two approaches at all three scales consistently and achieved the best accuracy with the error rate less than 2%.

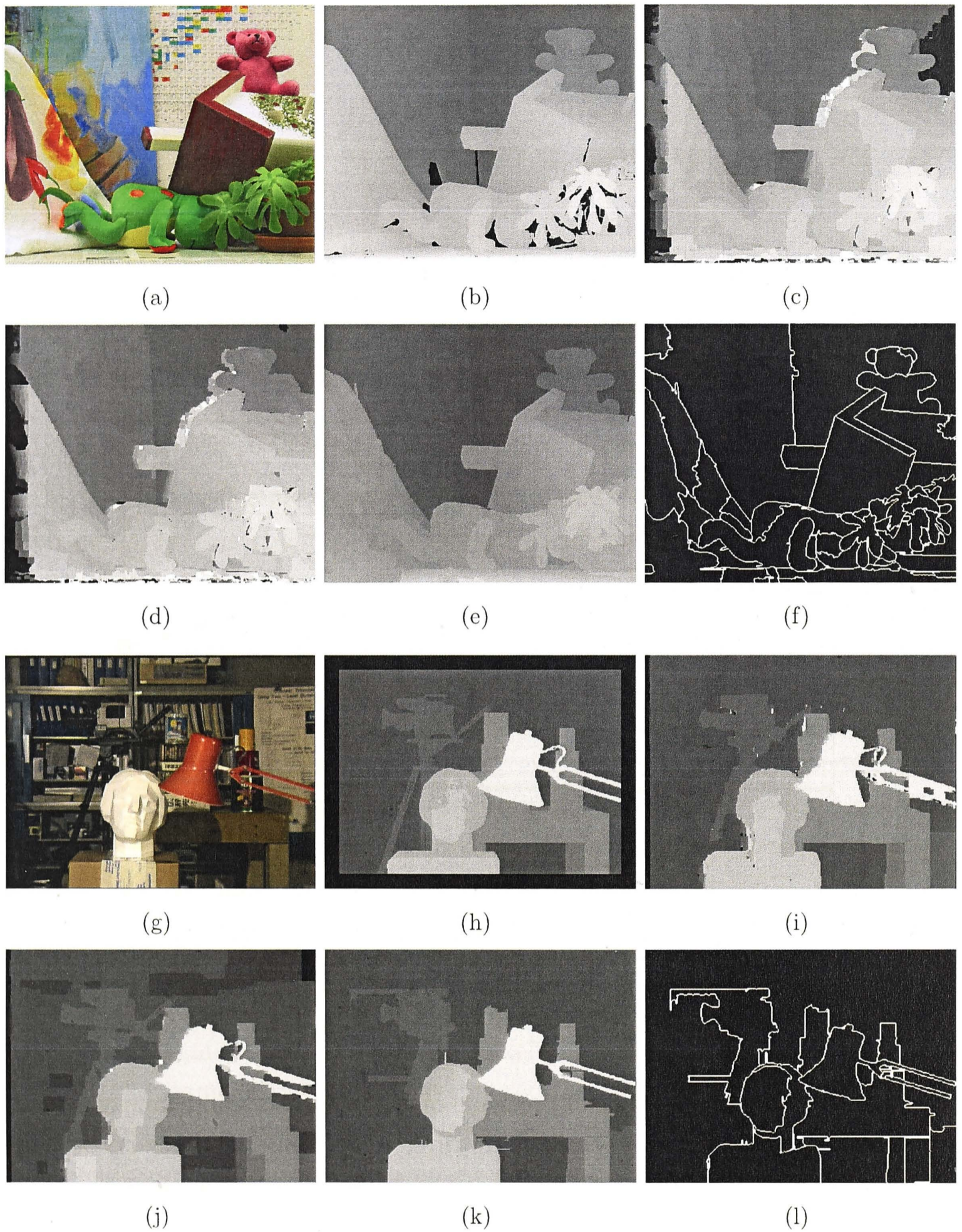


Figure 5.10: For quality review, we select to present the results on Teddy and Tsukuba from Middlebury. The first two rows are results on Teddy: (a) original image, (b) ground truth, (c) result by Graph Cuts, (d) result by Belief Propagation, (e) depth by proposed method, (f) surface boundary by proposed method. From (g) to (i) are results on Tsukuba.

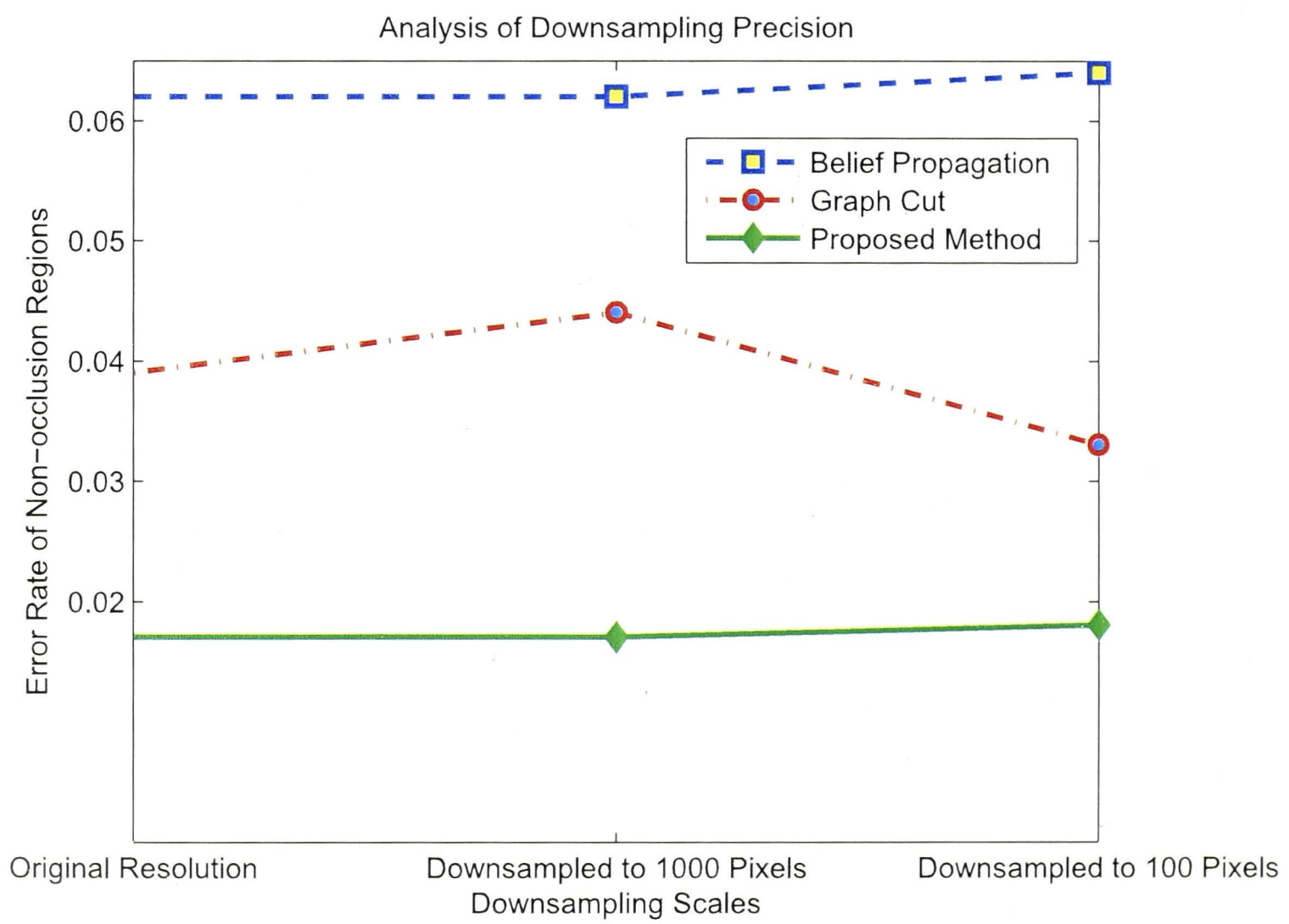


Figure 5.11: Quantity analysis of precision of the proposed algorithms comparing to Graph Cuts and Belief Propagation in three scales. The accuracy is computed as the average of Tsukuba, Teddy, Venus and Cones four image pairs.

Chapter 6

Conclusions

In this thesis, we have explored stereo matching problems using higher-order graph cuts. In this chapter, we summarize our contributions and state some future works.

6.1 Contributions

Firstly, we qualitatively and quantitatively evaluate five recently proposed state-of-the-art segmentation algorithms. In addition, we compare and analysis their performance in classic segment-based stereo matching algorithms. Through experiment, we conclude that color segmentation methods generally perform well because they are more coincident with the object boundaries. On the other hand, regularized segmentation algorithms are more suitable when the segmentation scale is large. The reason is the regularization on the size of segments makes every segment becomes a large “pixel”. This aims at helping researchers to choose the segmentation algorithm that most suitable for their stereo matching application.

Secondly, a novel approach to dense stereo matching has been provided. Conventional segment-based algorithms share a hard constraint that all pixels in the same segment must have the same depth value or lie on a locally fitted surface such as a plane, and discontinuities only occur on segment boundaries. While hard constraint helps reducing ambiguity of disparities, it is not robust. Different to theirs, our approach develops the idea of soft constraint that encourage but not force pixels to follow the same distribution if they are in the same segment. This idea has been transformed into a higher-order energy potential, and optimized along with unary and pairwise terms in our framework. Beyond the novel higher-

order term, the idea of sub-segmentation has been presented so that segments are not only decided by visual features but depth as well. For a better estimation, several successful techniques have been combined, including robust local matching method, left-right mutual check, confidence measurement, RANSAC and voting-based plane-fitting. Two test-beds of both Middlebury and challenging real-scene images have been evaluated, and results show that it obtains state-of-the-art results while still keeping efficiency.

Thirdly, we present a novel global optimization framework that combines stereo matching with surface boundary estimation. To encode the relationship between these two types of variables, a two-layer Markov Random Fields(MRFs) is built in which one layer represents depth and the other represents surface boundaries. In such framework, two types of variables are inferred globally and simultaneously. The work is carried out on both constrained Delaunay triangulation level and color segmentation level. The former one features depth boundary completion and the latter one provides accurate boundaries. We have successfully applied it in low-resolution phosphene-based indoor human navigation. With surface boundaries integration, it has three significant improvements:1) eliminating depth ambiguities and increasing the accuracy, 2) providing comprehensive information of depth and boundary for human navigation under low-resolution phosphene vision, 3) when integrating the boundary clues into downsampling process, the foreground obstacles are clearly enhanced and discriminated from the surrounding background. To optimize such complex graph, we choose the latest projected graph cuts. Experiments on both Middlebury and indoor real-scene data set show that the proposed approach achieves significantly better performance than other popular methods in both regular and low resolutions.

6.2 Future Works

Besides these contributions, we have great interests to extend our current work in the following directions:

6.2.1 Objects Recognition

In Chapter 5, we show that the proposed two-layer framework gives great estimations of both depth and surface boundary. One typical result is given in Figure 6.1, it can be observed that the obstacles are clearly distinguished from the surrounding background. This work can be further extended to recognition

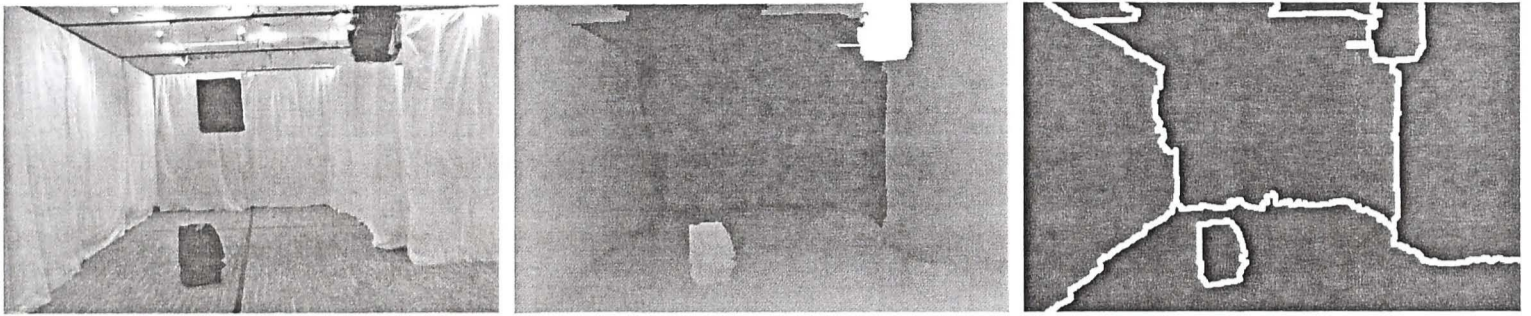


Figure 6.1: One of our results from two-layer framework described in Chapter 5. From left to right, input image with obstacles, our depth result, our surface boundary result.

and classification. Since objects have already been well discriminated, it will certainly assist future supervised learning work.

6.2.2 Hierarchical Model in Stereo Matching

Most of the existing works on stereo matching are either pixel-based or superpixel based, the former one gives a more precise estimation while the latter one is more efficient. To better find the trade off between accuracy and efficiency, a hierarchical model may be proposed. In some applications, eg., human navigation as we described in Chapter 5, the computation should focus on the close obstacles rather than the background objects which are far away. In other words, a rough depth estimation for background is sufficient, but for the nearer obstacles, the depth should be as accurate as possible. In a hierarchical model, every segment has its own children segments, the breaking down operation will dynamically happen only when the parent segment chooses certain depth labels. In our case, only if a segment takes a nearer depth label, its children nodes in the MRF will be visited. This tree model can be easily pre-computed in the initial color segmentation step with different color and space thresholds.

6.2.3 Projection Graph Cuts for Problems with Large Label Space

For labeling problems with large one-dimensional label space, conventional α -expansion will be inefficient, actually it takes a lot of time just visiting every label in the label set. To overcome it, the large label space can be decomposed into two dimensions, so the original variables will now be replaced by two new sets of variables. Once we have the labeling for both two new sets, the label assignment for the original variables can be easily computed through reverse conversion.

This method is in the hope of being more efficient while still achieving adequate accuracy.

In this case, any energy functions with second-order energy term will now be decomposed to a two-layer MRFs with the clique size up to 4, as shown in Figure 6.2. If the second-order term in the original energy function is in some specific forms, e.g., Potts model, the transformed two-layer structure can be solved through projection graph cut. In general cases, the fully linked clique can be approximated by 4 pairwise links through some techniques, e.g., least square error. Also if we force all weight to be positive, we will be able to solve two-layers together through max-flow algorithm.

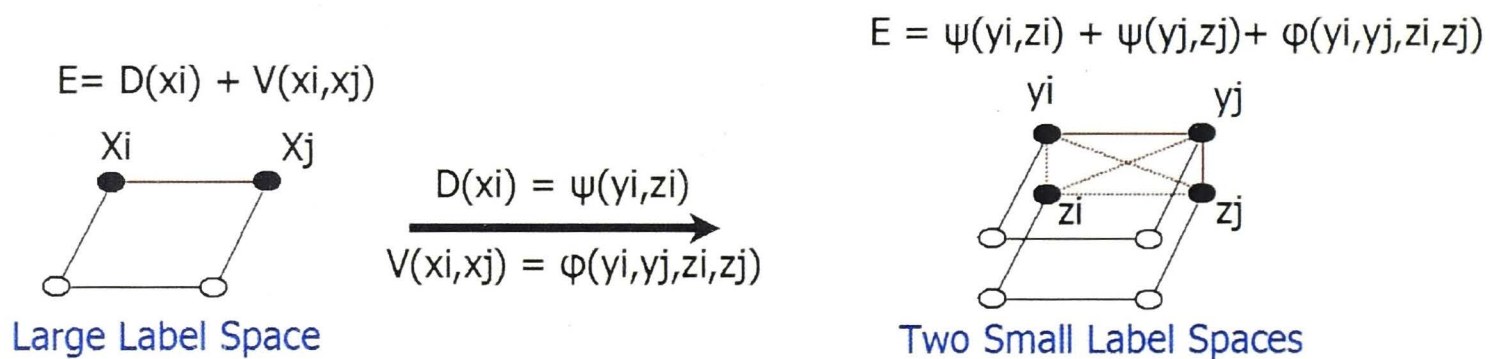


Figure 6.2: An illustration of label space decomposition.

Bibliography

- [1] P. Anandan. *A Computational Framework and an Algorithm for the Measurement of Visual*. University of Massachusetts, 1987.
- [2] R. D. Arnold. *Automated stereo perception*. PhD thesis, 1983.
- [3] S. Bagon, O. Boiman, and M. Irani. What is a good image segment? a unified approach to segment extraction. In *Proceedings of the European Conference on Computer Vision*, 2008.
- [4] S. T. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3:17–32, 1989.
- [5] S. T. Barnard and M. A. Fischler. Computational stereo. *ACM Computing Surveys*, 14:553–572, 1982.
- [6] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, B-48:259–302, 1986.
- [7] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35:269–293, 1999.
- [8] C. M. Bishop. *Pattern Recognition and Machine Learning, year = 2007*. Springer.
- [9] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *Proceedings of the Computer Vision and Pattern Recognition*, 2010.
- [10] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, 2004.
- [11] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 2001.

- [12] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24:325–376, 1992.
- [13] H. R. Carr, P. Minimizing energy functions on 4-connected lattices using elimination. In *Proceedings of the International Conference on Computer Vision*, 2009.
- [14] F. C.E.Liedtke, T.Gahm and B.Aeikens. Segmentation of microscopic cell scenes. *Anal Quant Cytol Histol*, 3:197–211, 1987.
- [15] P. K. P. T. Chris Russell, Lubor Ladicky. Exact and approximate inference in associative hierarchical random fields using graph-cuts. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2010.
- [16] D. Comaniciu, P. Meer, and S. Member. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.
- [17] A. I. H. B. Y. Delong, A. Osokin. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 1:1–27, 2010.
- [18] U. R. Dhond and J. K. Aggarwal. Structure from stereo-a review. *IEEE Transactions on Systems Man and Cybernetics*, 19:1489–1510, 1989.
- [19] G. Egnal and R. Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1127 – 1133, 2002.
- [20] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*.
- [21] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.
- [22] M. A. Fischler and R. C. Bolles. *Readings in computer vision: issues, problems, principles, and paradigms*. Morgan Kaufmann Publishers Inc., 1987.
- [23] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [24] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR : Proceedings of the Computer Vision and Pattern Recognition*, 2005.

- [25] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, 1986.
- [26] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. 1992.
- [27] S. Gould. Max-margin learning for lower linear envelope potentials in binary markov random fields. In *Proceedings of the International Conference on Machine Learning*, 2011.
- [28] P. B. Greig, D. and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.
- [29] M. J. Hannah. *Computer matching of areas in stereo images*. PhD thesis, 1974.
- [30] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *Proceedings of the Computer Vision and Pattern Recognition*, 2009.
- [31] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:920–932, 1994.
- [32] H. Kano. Development of a video-rate stereo machine. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 1995.
- [33] P. Kohli and M. P. Kumar. Energy minimization for linear envelope mrfs. In *Proceedings of the Computer Vision and Pattern Recognition*, 2010.
- [34] P. Kohli, M. P. Kumar, and P. H. S. Torr. P3 & beyond: Solving energies with higher order cliques. In *Proceedings of the Computer Vision and Pattern Recognition*, 2007.
- [35] P. Kohli, L. Ladický, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82:302–324, 2009.
- [36] P. Kohli and P. H. S. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *Proceedings of the International Conference on Computer Vision*, 2005.

- [37] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1568–1583, 2006.
- [38] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004.
- [39] C. R. S. S. Y. B. W. C. L. Ladicky, P. Sturges and P. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. In *Proceedings Of the British Machine Vision Conference*, 2011.
- [40] Lauritzen.S. *Graphical Models*. Oxford Science Publications, 1996.
- [41] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:1392–1405, 2010.
- [42] M. D. Levine and A. Nazif. Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:155–164, 1985.
- [43] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2290–2297, 2009.
- [44] S. Z. Li. *Markov Random Field Models in Computer Vision*. Springer, 1995.
- [45] P. Lieby, N. Barnes, C. McCarthy, N. Liu, L. Dennett, J. Walker, V. Botea, and A. Scott. Substituting depth for intensity and real-time phosphene rendering: Visual navigation under low vision conditions. In *Proceedings of the International Conference of the IEEE Engineering Medicine & Biology Society*. 2011.
- [46] D. Marr and T. Poggio. *Neurocomputing: foundations of research*. 1988.
- [47] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:530–549, 2004.

- [48] P. K. D. S. S. Michael Bleyer, Carsten Rother. Object stereo joint stereo matching and object segmentation. In *Proceedings of the Computer Vision and Pattern Recognition*, 2011.
- [49] A. P. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *Proceedings of the Computer Vision and Pattern Recognition*, 2008.
- [50] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:139–154, 1985.
- [51] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7:143–162, 1992.
- [52] N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer-Verlag New York, Inc., 2005.
- [53] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.
- [54] E. Pissaloux. A vision system design for blinds mobility assistance. In *Proceedings of the International Conference of the IEEE Engineering Medicine & Biology Society*, 2002.
- [55] R. Pissaloux, E. Velazquez and F. Maingreud. Intelligent glasses: A multi-modal interface for data communication to the visually impaired. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008.
- [56] A. P.K.Sahoo, S.Soltani and Y.C.Chen. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41:233–260, 1988.
- [57] T. Poggio. Early vision: From computational structure to algorithms and parallel hardware. *Computer Vision, Graphics, and Image Processing*, 31:139–155, 1985.
- [58] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Exact inference in multi-label crfs with higher order cliques. In *Proceedings of the Computer Vision and Pattern Recognition*, 2008.

- [59] X. Ren, C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. In *Proceedings of the International Conference on Computer Vision*, 2005.
- [60] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision*, 2003.
- [61] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast Cost-Volume Filtering for Visual Correspondence and Beyond. In *Proceedings of the Computer Vision and Pattern Recognition*, 2011.
- [62] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *Proceedings of the Computer Vision and Pattern Recognition*, 2009.
- [63] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. 2007.
- [64] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *Proceedings of the Computer Vision and Pattern Recognition*, 2005.
- [65] G. R. Ryan, T.W. and B. Hunt. Prediction of correlation errors in stereo-pair images. In *Proceedings of Optical Engineering*, 1980.
- [66] D. Scharstein. Matching images by comparing their gradient fields. In *Proceedings of the International Conference on Pattern Recognition*, 1994.
- [67] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2001.
- [68] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing: Analysis and Machine Vision*. CL-Engineering, 1998.
- [69] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, A. Agarwala, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. In *Proceedings of the European Conference on Computer Vision*, 2006.
- [70] L. N. B. N. Tong H, Liu S. A novel object-oriented stereo matching on multi-scale superpixels for low-resolution depth mapping. 2010.

- [71] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *Proceedings of the Computer Vision and Pattern Recognition*, 2005.
- [72] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51:2005, 2005.
- [73] Z. Wang and Z. Zheng. A region based stereo matching algorithm using cooperative optimization. In *Proceedings of the Computer Vision and Pattern Recognition*, 2008.
- [74] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.
- [75] W. T. M. S. M. Wentai Liu, Fink. Image processing and interface for retinal visual prostheses. In *Proceedings of the International Symposium on Circuits and Systems*, 2005.
- [76] G. Williams. *Linear Algebra With Applications*. Jones and Bartlett Publishers, Inc., 2007.
- [77] Q. Yang, C. Engels, and A. Akbarzadeh. Near real-time stereo for weakly-textured scenes. In *Proceedings Of the British Machine Vision Conference*, 2008.
- [78] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proceedings of the Computer Vision and Pattern Recognition*, 2006.
- [79] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, 2000.
- [80] J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Exploring artificial intelligence in the new millennium*. 2003.
- [81] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:650–654, 2006.

- [82] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision*, 1994.
- [83] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110:260–280, 2008.
- [84] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29:1335–1346, 1996.
- [85] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75:49–65, 2007.