Stereo Matching Based on Edge-Aware T-MST

by

Dan Zhou

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Dense stereo matching is one of the most extensively investigated topics in computer vision, since it plays an important role in many applications such as 3D scene reconstruction.

In this thesis, a novel dense stereo matching method is proposed based on edge-aware truncated minimum spanning tree (T-MST). Instead of employing non-local cost aggregation on traditional MST which is only generated from color differences of neighbouring pixels, a new tree structure, "Edge-Aware T-MST", is proposed to aggregate the cost according to the image texture. Specifically, cost aggregations are strongly enforced in large planar textureless regions due to the truncated edge weights. Meanwhile, the "edge fatten" effect is suppressed by employing a novel hybrid edge-prior which combines edge-prior and superpixel-prior to locate the potential disparity edges. Then a widely used *Winner-Takes-All* (WTA) strategy is performed to establish initial disparity map. An adaptive non-local refinement is also performed based on the stability of initial disparity estimation.

Given the stereo images from Middlebury benchmark, we estimate the disparity maps by using our proposed method and other five state-of-the-art tree-based non-local matching methods. The experimental results show that the proposed method successfully produced reliable disparity values within large planar textureless regions and around object disparity boundaries. Performance comparisons demonstrate that our proposed non-local stereo matching method based on edge-aware T-MST outperforms current non-local tree-based state-of-the-art stereo matching methods in most cases, especially in large textureless planar regions and around disparity bounaries.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my thesis supervisor Professor Jiying Zhao for his patience, motivation, enthusiasm, and immense knowledge, as well as the continuous support that he provided me throughout this entire research endeavour.

I would also like to thank all my lab colleagues: Wenyi Wang, Lei Chen, Jun Hu, Ya Luo, Mengdie Chu, for their generous help and support during the last two years.

My sincere thanks also goes to my boyfriend Xiaohong Liu, for his generosity, patience, optimism and love.

Last but not least, my special thanks goes to my family, for their constant love, encouragement and support.

This thesis is dedicated to my family.

Table of contents

List of tables

List of figures

Nomenclature

Abbreviations

AD	Absolute Intensity Differences
BPP	Bad Pixel Percentage
CC	Cross-Correlation
CRF	Conditional Random Field
CT	Census Transform
DP	Dynamic Programming
HOG	Histogram of Oriented Gradient
MRF	Markov Random Field
MST	Minimum Spanning Tree
NCC	Normalized Cross-Correlation
RMS	Root-Mean-Squared
SAD	Sum of Absolute Differences
SD	Squared Intensity Differences
SGM	Semi-Global Matching
SP	Superpixel

SSD	Sum of Squared Differences
ST	Segment-Tree
T-MST	Truncated Minimum Spanning Tree
TAD	Truncated Absolute Intensity Differences
WTA	Winner-Takes-All

Chapter 1

Introduction

1.1 Stereo vision

Binocular vision [1], which uses two eyes with overlapping fields of views, allows good perception of depth according to the difference between the two similar views from left and right eyes (Figure 1.1). Intuitively, objects closer to viewer have larger displacement between left-eye view and right-eye view than those that are further away. This visual disparity provides guidance on how to understand, respond and interact with our surroundings.

Stereo vision, which simulates the similar behaviour to binocular vision, is still challenging after it has been extensively investigated for many years. In 1838, the concept stereopsis was firstly proposed by Charles Wheatstone [2]. In Victoria time, the invention of the prism stereoscope by David Brewster made stereoscopy popular in public. In the 1960's, scientists started their research on stereopsis to find its limitations and its relationship to monocular vision.

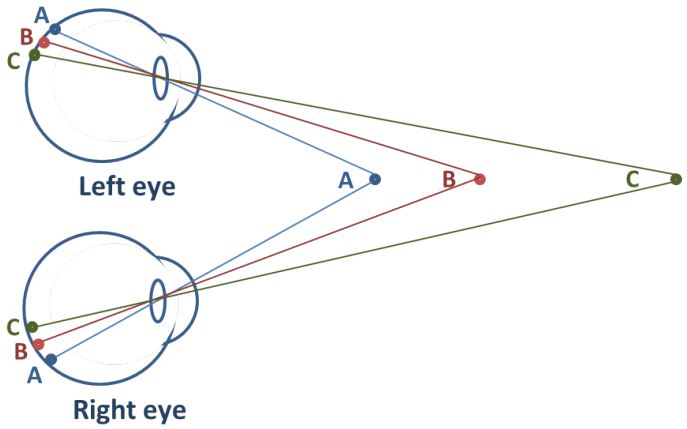


Figure 1.1: Binocular vision.

In the stereo vision system, the left-eye image and the right-eye image are obtained by two cameras simultaneously. The displacement of the same object (disparity) between the image pair can be transformed into the distance from the object to the viewer (i.e., depth).

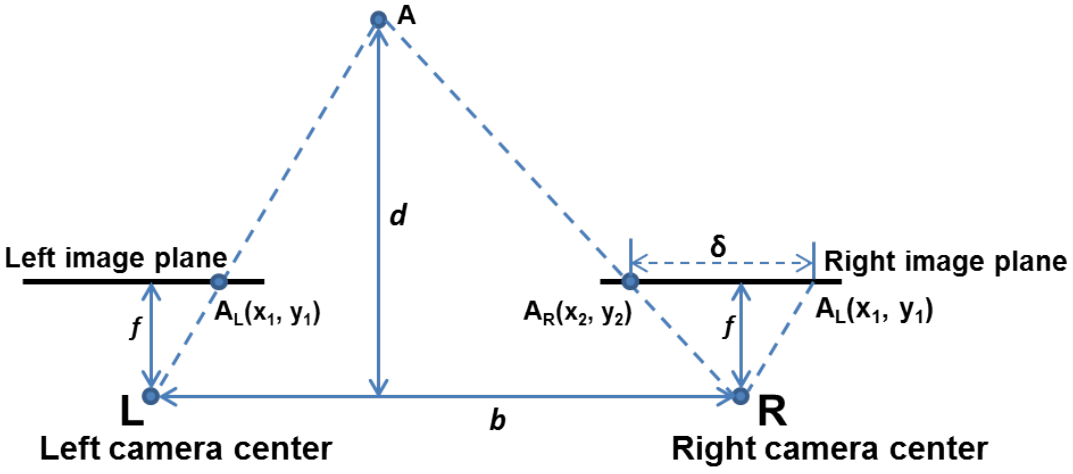


Figure 1.2: Stereo vision.

Figure 1.2 illustrates how to calculate depth d in the stereo vision system. L and

R represent the imaging centres of the left camera and the right camera respectively. The line connecting two camera imaging centres is referred as the baseline, and b represents the length of the baseline in pixel. Assume a point A is being viewed by both cameras, the projection of A in the left image plane is A_L at location (x_1, y_1); the projection of A in the right image plane is A_R with location (x_2, y_2). If we assume that the left image and the right image are rectified, corresponding points should lie on the same epipolar line ($y_1 = y_2$). The disparity δ between A_L and its corresponding point A_R can be calculated by $\delta = |x_1 - x_2|$. The depth d of A in 3D space can be calculated by Equ. (1.1):

$$d = f \cdot \frac{b}{\delta}\,, \tag{1.1}$$

where f represents the focal length of the cameras.

In the stereo matching system, a disparity map is an image with each pixel value indicating the difference between the horizontal coordinate of a pair of corresponding pixels from the stereo image pairs, while a depth map is an image containing information relating to the distance between a viewpoint and the surface of scene objects. In recent research, estimating the disparity map or depth map from stereo image pair has attracted more interest than simulating 3D scene directly from stereo image pair.

The depth information can be obtained by some devices directly, such as laser sensor, infra-red ray sensor and light pattern sensor, or by using the image processing technologies. Image processing technologies for depth estimation are mainly based on monocular extraction or binocular extraction. Figure 1.3 shows the classification of depth cues.

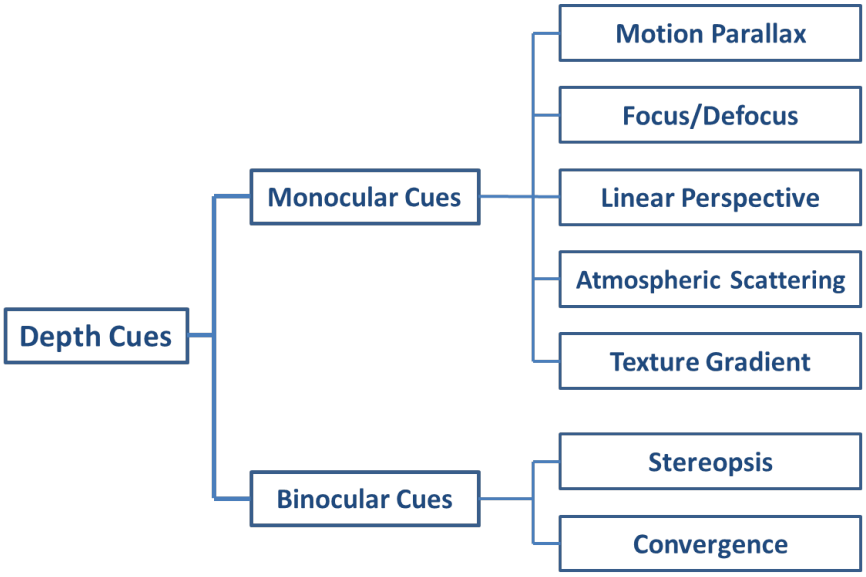


Figure 1.3: Classification of depth cues.

1.2 Stereo matching applications and methods

Stereo matching, which is one of the most compelling topics in computer vision, is a process of finding corresponding pixels given a set of images taken from different viewpoints.

Early applications of stereo matching was in photogrammetry [3, 4]. Given a set of calibrated images, the aim is to measure the structure of an object surface. For example, the topographic maps can be generated from satellite pictures. Before the automatic image processing technologies were proposed, device known as stereo plotter was used for manual stereo matching [5].

Nowadays, stereo matching can be widely applied in entertainment (e.g., gesture capturing by Kinect through stereo cameras), information transfer (e.g., 3D scene reconstruction [6], intermediate view creation [7]) and automated systems (e.g.,

anonymous driving [8] and robotics [9]).

In recent years, a large variety of stereo matching algorithms have been developed to determine the disparities indicating the horizontal difference of the corresponding pixels. Ideally, we assume that the corresponding pixels have the same intensity value when solving the stereo matching problem. In practice, however, stereo images always suffer from several problems, such as image noise, illumination variation, specular reflection and object transparency. In order to make stereo matching problem analysable and solvable, several general assumptions are made as follows [10]:

1. Lambertian surface: The apparent brightness of a Lambertian surface does not vary regardless of the observer's viewpoint.

2. Epipolar constraint: For each pixel in the image, the corresponding pixel in the other image must lie on a known epipolar line. Particularly, when dealing with calibrated stereo image pairs, we only search along the epipolar line, which means a pair of corresponding pixels have the same y coordinate.

3. Continuity constraint: Disparity tends to change slowly and smoothly across a surface. If two points locate closely in one image, their corresponding pixels in the other image should also be close to each other.

4. Smoothness constraint: Disparity value in a given neighbourhood should be the same (or similar), except for depth boundaries.

5. Ordering constraint: The relative position between two pixels in one image should be preserved in the other image for their corresponding pixels.

6. Uniqueness constraint: For each pixel in one image, there should be at most one corresponding pixel in the other image.

7. Maximum disparity constraint: A probable maximum disparity is computed to be used as searching range for every stereo image pair.

However, there are still some challenging problems for stereo matching algorithms in practice:

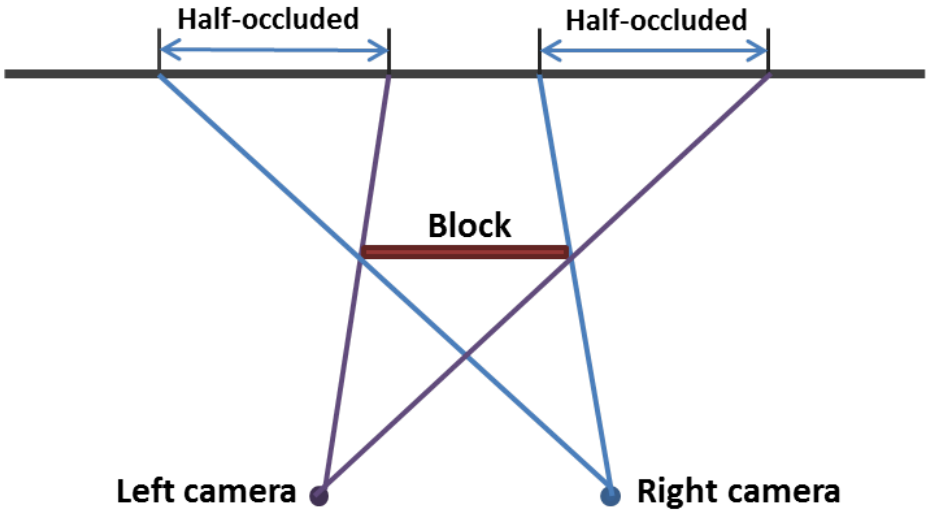


Figure 1.4: The occlusion problem.

1. Occlusion problem: Occlusion problem is a very typical problem in stereo matching since it is difficult to find corresponding pixels when one of them is hidden or does not exist. An illustration of occlusion is shown in Figure 1.4, where the areas marked as "half-occluded" can only be viewed with one camera due to the block.

2. Non-Lambertian surface: In real world, the Lambertian surface assumption is not always true. A change of viewing angle may cause drastically different intensities for a pair of corresponding pixels in different images.

3. Large textureless region: It is impossible to uniquely identify the corresponding pixels in those regions due to the lack of texture information.

With these challenges, it is important for stereo matching algorithms to overcome those problems and generate an accurate disparity map.

The goal of this thesis, in addition to overcome the aforemetioned general problems and challenges, is to design a stereo matching scheme which balances the speed and accuracy.

1.3 Contributions

In this thesis, we propose a novel edge-aware truncated minimum spanning tree (T-MST) structure for performing non-local cost aggregation. Different from traditional local window-based algorithms which only analyse the pixel within a fixed support window, the pixels can receive weighted supports from the entire image in non-local algorithms that generate a tree structure for all the pixels in the image. In our algorithm, the cost aggregation in highly textured regions is promoted by assigning truncated weights to edges in T-MST. Meanwhile, a hybrid edge prior which combines the edge prior and the superpixel prior is also proposed to restrain the "edge fatten" effect across disparity boundaries.

A novel matching cost function is also proposed in this thesis. This matching

cost function is a convex combination of truncated absolute differences (TAD) of both intensity information and gradient information, and an improved histogram of oriented gradient (HOG) feature. This function is robust to outlier pixels, vertical parallax and illumination variation.

In addition, a novel adaptive non-local refinement algorithm is proposed to make full use of the stability information obtained from left-right consistency check. The cost aggregations from unstable pixels to stable pixels are suppressed to improve the accuracy of the refined disparity map.

The proposed method calculates the disparity maps of the stereo images in Middlebury benchmark. The experimental results show that our proposed algorithm outperforms the current state-of-the-art non-local tree-based stereo matching algorithms.

1.4 Thesis structure

In Chapter 2, related fundamental concepts and techniques behind stereo matching are introduced. Different algorithms for solving correspondence problem are presented, followed with introduction of basic matching cost functions. Moreover, concepts of graph and minimum spanning tree in the field of image processing are described. In addition, the evaluation scheme for dense stereo matching algorithms proposed by Scharstein and Szeliski [10] is also presented.

Chapter 3 gives the literature review of existing dense stereo matching algorithms. Local window-based algorithms and global energy minimization algorithms are two broad categories in this field. Local algorithms are fast but less accurate than global

algorithms while global algorithms produce accurate disparity maps at the cost of much higher computational complexity. Several current state-of-the-art non-local stereo matching algorithms, which balance the efficiency and accuracy, are then presented in this chapter.

Chapter 4 presents our proposed novel non-local cost aggregation algorithm over an edge-aware T-MST. A hybrid edge prior which combines the edge prior and superpixel prior is proposed for non-local cost aggregation on T-MST. In addition, a matching cost function which is robust to illumination variation and a novel non-local refinement method which takes advantage of pixel stabilities are also proposed in our non-local stereo matching framework.

In Chapter 5, the experimental results of our proposed method are presented. Comparisons of the computed disparity maps on 30 Middlebury data sets by our algorithm and by the five state-of-the-art non-local tree-based algorithms show that the proposed algorithm has the best overall accuracy and ranking.

Chapter 6 draws the conclusions of this thesis.

Chapter 2

Fundamental concepts and techniques

In this chapter, we introduce the fundamental concepts and techniques behind stereo matching. First, we review the correspondence problem and the basic matching cost functions that find the corresponding points between a pair of stereo images. Second, we study the concepts and applications of the graph theory and the minimum spanning tree (MST). Last but not least, we review the evaluation scheme of stereo matching algorithms.

2.1 Correspondence problem

The correspondence problem refers to the task of matching a set of points with their corresponding points given two or more images of the same 3D scene taken from different viewpoints [11]. Numerous algorithms have been proposed to solve the cor-

respondence problem over the years. These algorithms can be divided into two broad categories: sparse correspondence and dense correspondence.

Sparse correspondence methods match the features of interest extracted from the images such as edges and contours. These methods are fast since only a small subset of pixels are used for matching. Interpolation techniques are needed to fill the sparse disparity maps generated by this type of methods.

Dense correspondence methods, on the other hand, seek to find the correspondence for each pixel in the image since a smooth and detailed disparity map is more useful for subsequent 3D modelling and rendering. According to the review of dense correspondence algorithms in [10], dense disparity maps can be computed and refined by calculating and aggregating matching costs. Dense stereo matching algorithms can be subdivided into two main categories: local algorithms and global algorithms. In this thesis, the details of dense stereo matching algorithms will be introduced in Chapter 3.

Regardless of the type of stereo matching algorithm being used, a proper means to determine the similarity between pixels in stereo images is crucial for finding the correct corresponding pixels. Given a pixel p in the left image I_l with coordinate (x, y), the corresponding pixel at disparity d in the right image I_r is denoted as p_d with coordinate $(x - d, y)$.

Figure 2.1 illustrates how to find the corresponding pixel p_d in the right image I_r given a pixel p in the left image I_l. The blue dots represent the pixels in the image and the red dots represent two corresponding pixels p and p_d with the disparity of 2.

Commonly-used pixel-based matching cost functions are the absolute intensity differences (AD) and the squared intensity differences (SD), which are presented as

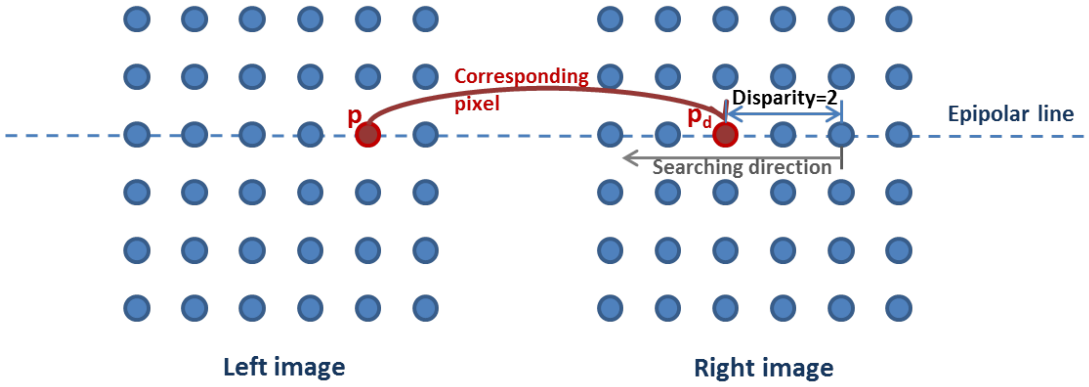


Figure 2.1: Illustration of finding corresponding pixel of p.

follows:

$$C_d^{AD}(p) = |I_l(p) - I_r(p_d)|\,, \tag{2.1}$$

$$C_d^{SD}(p) = [I_l(p) - I_r(p_d)]^2\,, \tag{2.2}$$

It is obvious that AD and SD are not robust since they are very sensitive to random image noise. Another pixel-based cost function is the truncated absolute intensity differences (TAD), as shown in Equ. (4.2):

$$C_d^{TAD}(p) = min(|I_l(p) - I_r(p_d)|, \tau)\,, \tag{2.3}$$

where τ is the truncation threshold that helps reduce the influence of outliers.

Compared with pixel-based matching cost functions, window-based cost functions are more robust. A fixed-size support window centred at pixel p is used to measure the

similarity of its corresponding pixel p_d in the other image. A basic window-based cost function is the sum-of-squared-differences (SSD) function, as shown in Equ. (2.4):

$$C_d^{SSD}(p) = \sum_{i \in W} [I_l(p^i) - I_r(p_d^i)]^2 \,, \tag{2.4}$$

where p^i denotes a pixel within the support window W and p_d^i denotes the corresponding pixel at disparity d.

The SSD function implicitly assumes that the corresponding pixels in stereo images have the same intensity value. However, when the support window covers outliers such as image noise, SSD cost grows rapidly. Another common-used window-based matching cost function is the sum-of-absolute-differences (SAD) function, as shown in Equ. (2.5):

$$C_d^{SAD}(p) = \sum_{i \in W} |I_l(p^i) - I_r(p_d^i)| \,. \tag{2.5}$$

The SAD function grows linearly with the residual error, thus the influence of mismatches can be reduced during aggregation step. Truncated quadratics and contaminated Gaussians are also robust matching cost functions [10].

In practice, it is common that the two stereo images are taken at different exposures. In this case, matching cost functions that are invariant to inter-image intensity differences are needed. A simple model of linear intensity variation between a pair of images is presented as follows:

$$I_l(p) = (1 + \alpha) I_r(p_d) + \beta \,, \tag{2.6}$$

where α denotes the gain and β denotes the bias. Hence, matching cost functions can be modified to take intensity variations into account at the cost of higher computational complexity. For instance, the SSD function can be rewritten by Equ. (2.7):

$$C_d^{SSD}(p) = \sum_{i \in W} [I_l(p^i) - (1+\alpha) I_r(p_d^i) - \beta]^2 \,. \tag{2.7}$$

The Census Transform (CT) is another window-based matching cost function with high robustness to exposure and illumination variation [12, 13]. Different from other matching cost functions that directly rely on the intensity values, the CT relies on the ordering of relative pixel intensities [14]. The CT function is defined by Equ. (2.8):

$$C_d^{CT}(p) = hdist(v(p), v(p_d)) \,, \tag{2.8}$$

where *hdist* denotes the Hamming distance of the two vectors; $v(p)$ and $v(p_d)$ denote the CT bit-vectors of the corresponding pixels within support windows using a comparison function in Equ. (2.9):

$$\xi(p, p^i) = \begin{cases} 0 & p \leqslant p^i \,, \\ 1 & p > p^i \,. \end{cases} \tag{2.9}$$

The illustration of calculating CT cost of a given pixel p at disparity d is shown in Figure 2.2. In Figure 2.2, the intensity of p is 110. A bit-vector is generated based on the intensities of eight neighbouring pixels of p. If the intensity of p is larger than the intensity of its neighbour, the corresponding bit is set to 1. Otherwise, the bit is set to 0. The order of each bit in the CT bit-vector is shown in Figure 2.2 with

arrows. The CT cost of p and p_d is the Hamming distance of the two vectors.

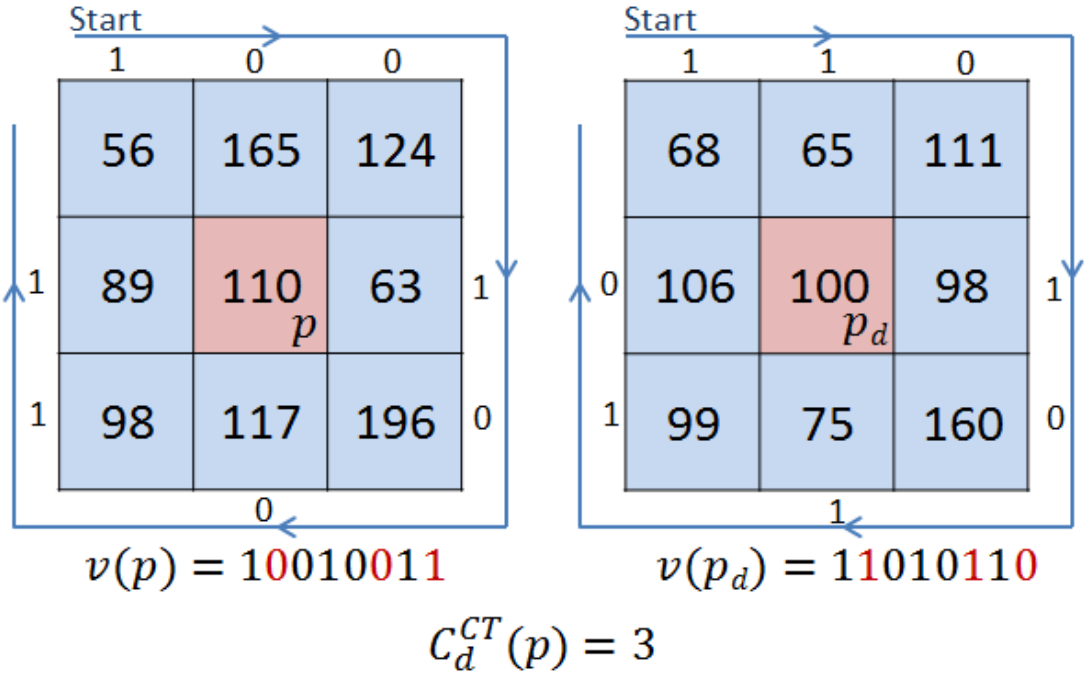


Figure 2.2: Illustration of CT function.

Alternatively, matching cost functions using gradient values instead of intensity values was proved to be robust to exposure variation and illumination variation [15, 16]. Another approach is to subtract the window average, which has also been shown to be robust [17, 18].

Besides the aforementioned window-based matching cost functions that measure the residual error, another commonly-used cost function is the cross-correlation (CC) function, as shown in Equ. (2.10):

$$C_d^{CC}(p) = \sum_{i \in W} I_l(p^i) I_r(p_d^i)\,. \tag{2.10}$$

Different from cost functions for which the most probable match among all possible disparity levels is the one with minimum cost, the maximum CC cost corresponds to the best match. However, the CC function can fail when the images have a large dynamic range or the illumination condition changes across images. In order to avoid those problems, the normalized cross-correlation (NCC) function can be employed, as shown in Equ. (2.11):

$$C_d^{NCC}(p) = \frac{\sum_{i \in W}[I_l(p^i) - \overline{I_l(p^i)}][I_r(p_d^i) - \overline{I_r(p_d^i)}]}{\sqrt{\sum_{i \in W}[I_l(p^i) - \overline{I_l(p^i)}]^2[I_r(p_d^i) - \overline{I_r(p_d^i)}]^2}}, \tag{2.11}$$

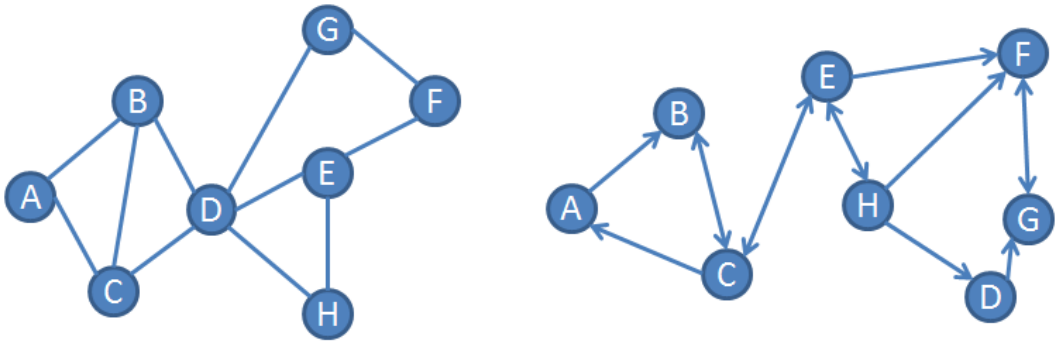
where $\overline{I_l(p^i)}$ and $\overline{I_r(p_d^i)}$ denote the mean intensities of the corresponding support windows. A value of 1 in NCC cost indicates the perfect match.

2.2 The graph theory

In computer science, a graph is an abstract data type that implements the concepts of undirected graph and directed graph.

A graph contains a finite set of vertices (also called nodes or points) and a set of edges connecting pairs of these vertices. In an undirected graph, these edges are also known as arcs or lines; in a directed graph, these edges can be called as arrows, directed edges, directed arcs, or directed lines. In graph structure, weighted value can be assigned to each edge, such as a symbolic label or a numeric attribute. Figure 2.3 (a) shows an example of an undirected graph; Figure 2.3 (b) shows an example of a directed graph.

A large variety of problems in computer science can be modelled by graph model.



(a) Undirected graph (b) Directed graph

Figure 2.3: Examples of graph.

For instance, the famous travelling salesman problem can be modelled as an undirected weighted graph, in which the vertices are the cities; the edges are the paths between cities and the each edge weight represents the path's distance.

Graph can also be applied in image processing techniques since an image I can be represented as a 4-connected undirected graph $G(V, E)$ [19, 20]. The vertices V are the pixels in the image I and the edges E are the connecting edges between neighbouring pixel pairs. An edge weight function ω is defined to map edges E with real-valued weights based on the pixel intensities.

A simple weight function is shown in Equ. (2.12) [19]:

$$\omega(s, r) = \omega(r, s) = |I(s) - I(r)|\,, \tag{2.12}$$

where s and r denote a pair of neighbouring pixels in image I.

Figure 2.4 shows an illustration of a 4-connected undirected graph generated from a 5×5 image. In this example, the blue circles represent pixels in the image; the

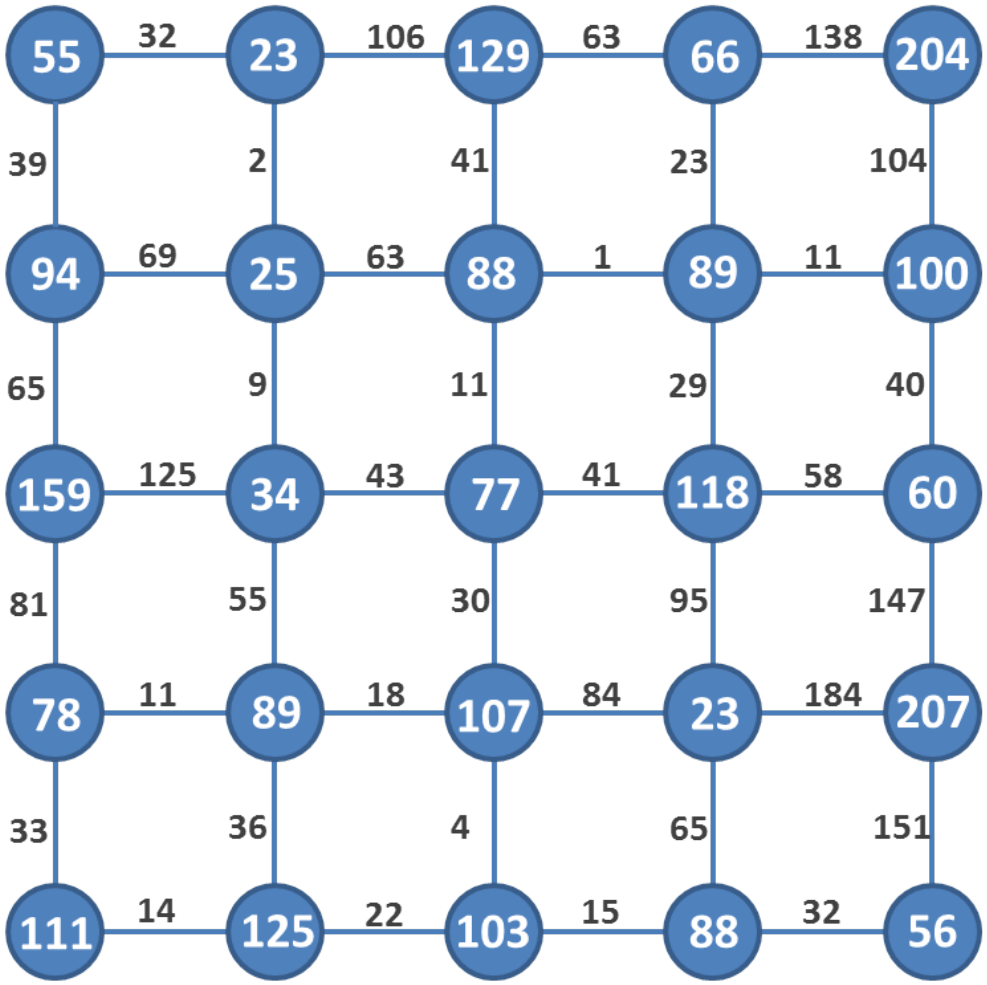


Figure 2.4: Illustration of a 4-connected undirected graph generated from an image.

white numbers inside the circles represent the corresponding pixel intensities; the lines represent the edges connecting neighbouring pixels; the black numbers beside the lines represent the corresponding edge weights.

Representing image in the form of graph structure allows neighbouring pixels to share information iteratively or in sequence [21]. In the area of stereo matching, a variety of algorithms have been proposed based on cost aggregation over graphs generated from images. These algorithms include graph cuts [22–24], dynamic programming (DP) [25, 26], belief propagation [27–29], tree-reweighted message passing [30, 31], and semi-global matching (SGM) [32, 33].

In order to improve the cost aggregation effectively and efficiently, tree structure spanning the whole image can be employed instead of 4-connected graph structure.

If we assign an edge weight to every edge in a connected, undirected graph, different spanning trees can be generated. The weight of a spanning tree is calculated by summing all the edge weights in this spanning tree. A minimum spanning tree (MST) is then a spanning tree with the minimum weight among all the spanning trees generated from the same graph.

There are four classic greedy algorithms developed for finding MST:

1. Boruvka's algorithm: This algorithm can only used for finding MST in a graph in which all edge weights are distinct. The computational complexity of Boruvka's algorithm is $O(MlogN)$, where M and N denote the number of edges and vertices in the graph respectively;

2. Prim's algorithm: This algorithm starts with an arbitrary vertex and adds the cheapest possible connection to another vertex repeatedly until all the vertices

are in this MST. The computational complexity of Prim's algorithm is either $O(MlogN)$ or $O(M + NlogN)$, depending on the data-structure;

3. Kruskal's algorithm: This algorithm builds the MST by iteratively removing the edges with large weights. The computational complexity of Kruskal's algorithm is $O(MlogN)$;

4. Reverse-delete algorithm: This algorithm is the reverse of Kruskal's algorithm with computational complexity $O(MlogN(loglogN)^3)$.

Figure 2.5 shows the MST generated from the graph shown in Figure 2.4 using Kruskal's algorithm. The red lines denote the edges in this MST.

MSTs can be used in many applications, such as the design of networks (computer networks, transportation networks, telecommunication networks), cluster analysis, taxonomy, image registration [34], image segmentation [35], and stereo matching [19].

In this thesis, we deal with the dense stereo matching problem by performing efficient cost aggregation over MST.

2.3 Evaluation of dense stereo matching methods

With the development of a large variety of stereo matching algorithms, quantitative evaluation method is needed for estimating the quality of the computed disparity maps. There are two general approaches: calculating the error between the computed disparity map and the ground truth [36] or evaluating the warped image generated from the reference image and the computed disparity map [37].

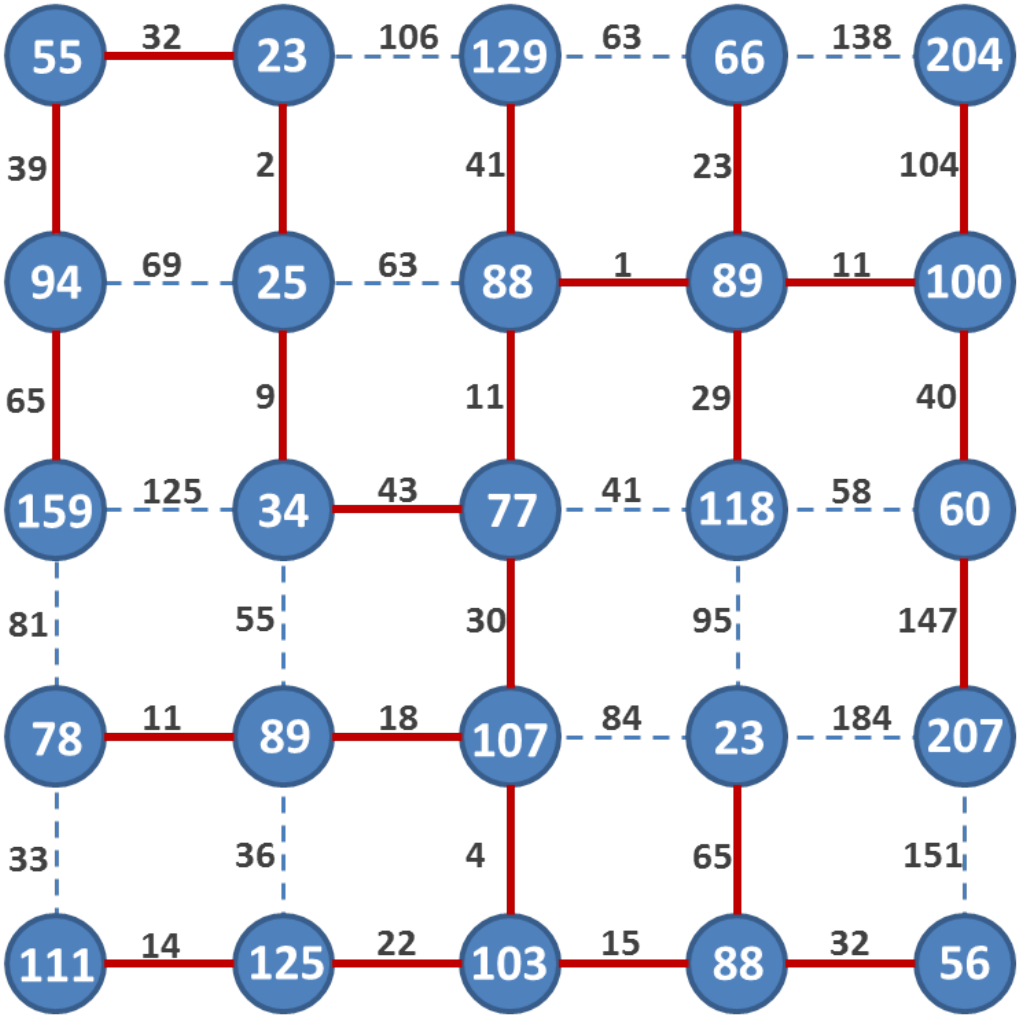


Figure 2.5: The MST generated from a 4-connected undirected graph.

According to the taxonomy and evaluation scheme proposed by Scharstein and Szeliski [10], two quality measurements can be used for stereo matching algorithm evaluation based on known ground truth.

1. Root-mean-squared (RMS) error:

$$R = \left(\frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_{GT}(x,y)|^2 \right)^{\frac{1}{2}}, \tag{2.13}$$

   where d_C and d_{GT} represent the computed disparity map and the ground truth map respectively; N denotes the total number of pixels in the image.

2. Bad pixel percentage (BPP):

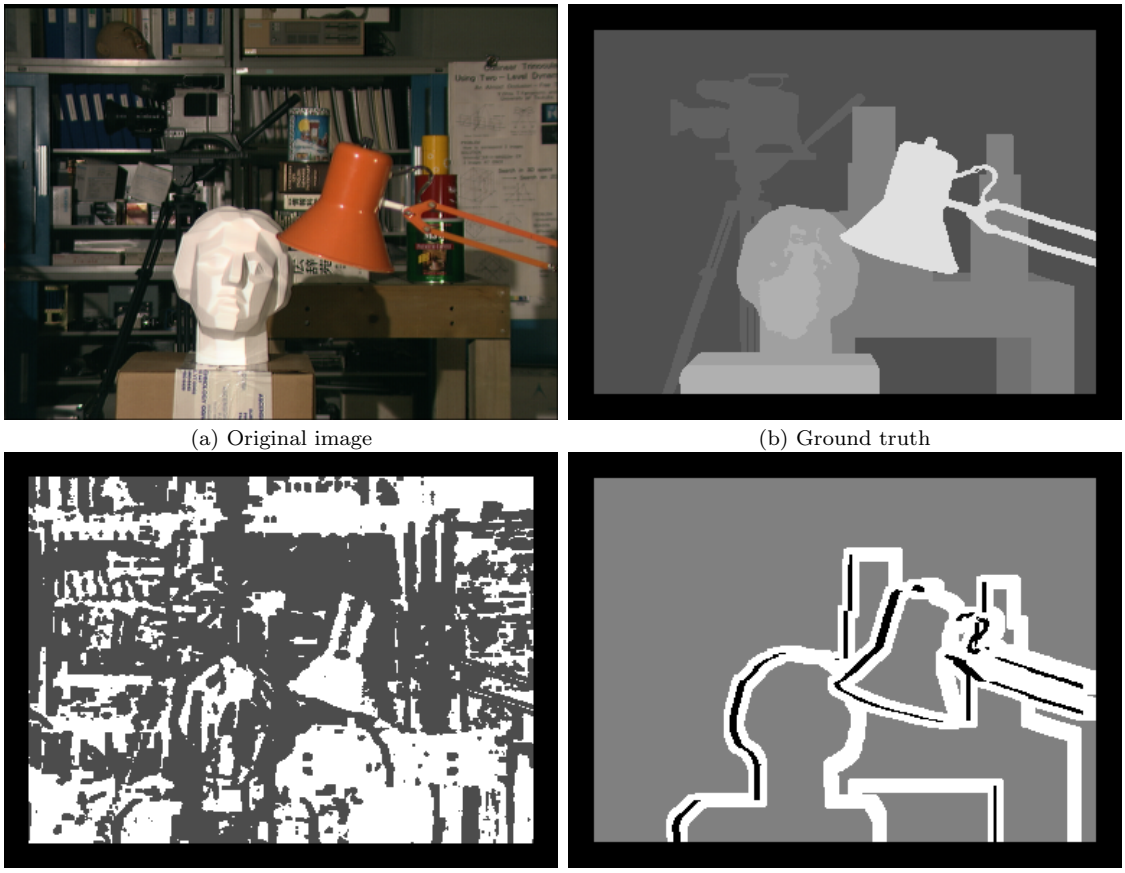
$$B = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_{GT}(x,y)| > \delta_d), \tag{2.14}$$

   where δ_d is the tolerance of disparity error.

In order to statistically evaluate the stereo matching algorithms from different aspects, three kinds of regions are considered [10]:

- Textureless regions T: regions with low average horizontal intensity gradient, as shown in white regions in Figure 2.6 (c);
- Occluded regions O: regions that are occluded in the original stereo images, as shown in black regions in Figure 2.6 (d). Occluded regions can be obtained by performing left-right consistency check on both ground truth maps.

- Depth discontinuity regions D: regions with pixels whose disparities differ from neighbouring pixels greatly (each pixel is enlarged with a small window), as shown in white regions in Figure 2.6 (d).



(a) Original image

(b) Ground truth

(c) Textureless regions (white)

(d) Occluded regions (black) and depth discontinuity regions (white)

Figure 2.6: Segmented region maps.

By using corresponding range of pixels, the BPP in different regions can then be calculated. For instance, the BPP in textureless regions T can be defined as below:

$$B_T = \frac{1}{N_T} \sum_{(x,y)\in T} \left(|d_C(x,y) - d_{GT}(x,y)| > \delta_d \right), \tag{2.15}$$

where N_T denotes the total number of pixels in regions T.

2.4 Summary

In this chapter, the fundamental concepts and basic techniques about stereo matching were introduced. The correspondence problem and common matching cost functions for finding corresponding points between stereo image pairs were presented. Then the concepts of the graph theory and MST were introduced. At last, the evaluation techniques for stereo matching algorithms were reviewed.

Chapter 3

Literature review

3.1 Local and global methods

Dense stereo matching is one of the most extensively investigated topics in computer vision since it plays an important role in a large variety of applications such as 3D scene reconstruction [6], intermediate view creation [7], anonymous driving [8] and robotics [9]. According to the taxonomy and evaluation scheme proposed in [10], stereo matching algorithms can be divided into two categories: local algorithms (window-based algorithms) and global algorithms (energy minimization algorithms). Stereo matching algorithms generally implement a subset of the following four steps:

1. Matching cost estimation;
2. Cost aggregation within support region;
3. Disparity computation/optimization;

4. Disparity refinement.

Local algorithms generally employ steps 1, 2 and 3 while global algorithms perform steps 1, 3 and 4. Matching cost (step 1) is computed firstly for both local and global methods. In local algorithms, costs for each pixel in different disparity levels are then aggregated (step 2) within its support region (usually a window). On the other hand, global algorithms make explicit smoothness assumptions and minimize a global energy function (step 3). In step 4, post-processing techniques are performed on the disparity maps to achieve a better result. Global methods usually generate more accurate results than the local methods do. However, the quality improvement is achieved at the cost of expensive computation.

3.1.1 Local algorithms

For traditional local stereo matching algorithms, the matching costs for each pixel in different disparity levels are aggregated within a support window as shown in Figure 3.1.

Since each pixel can only receive supports from pixels within the fixed support window, it is crucial to select an appropriate window size so that the final disparity map can be smooth and accurate. A proper support window should be large enough to contain enough information for reliable matching. If the window size is too small, it may be difficult to uniquely identify the correct matching pixel. Meanwhile, the support window should also be small enough to avoid covering disparity discontinuities. If the window size is too large, it may lead to inappropriate smoothing and cause “edge fatten” effect around disparity boundaries. Therefore, researchers have

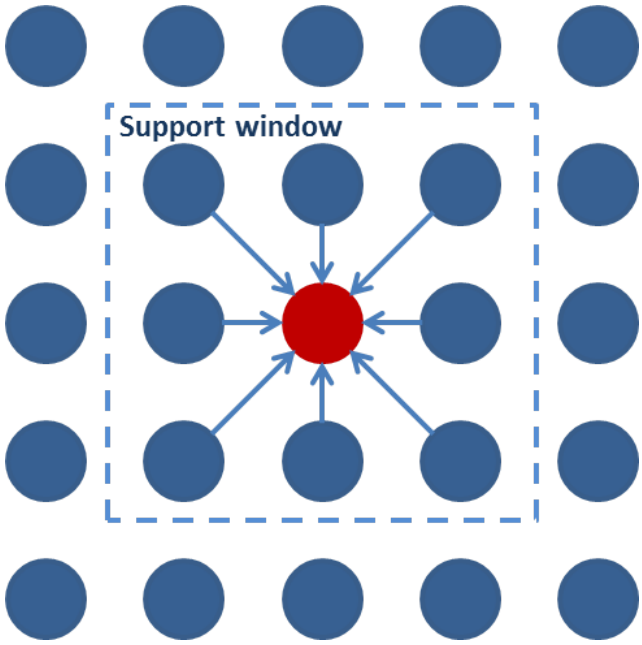


Figure 3.1: Local aggregation within a support window.

been seeking better techniques to balance the trade-offs involved in window selection rather than using a fixed window. In general, these algorithms employ adaptive support which dynamically changes according to the surroundings of each pixel. Such methods include adaptive window (sizes and shapes) [17, 38, 39], shiftable window [40, 41], and adaptive weights based on segmentation [42, 43].

After aggregating the matching costs, the most possible disparity for each pixel can be chosen by applying the Winner-Takes-All (WTA) strategy. Figure 3.2 shows an example of choosing the most possible disparity of pixel p given its matching costs at each disparity levels.

A disparity map can be generated by repeating the WTA strategy for every pixel in the image.

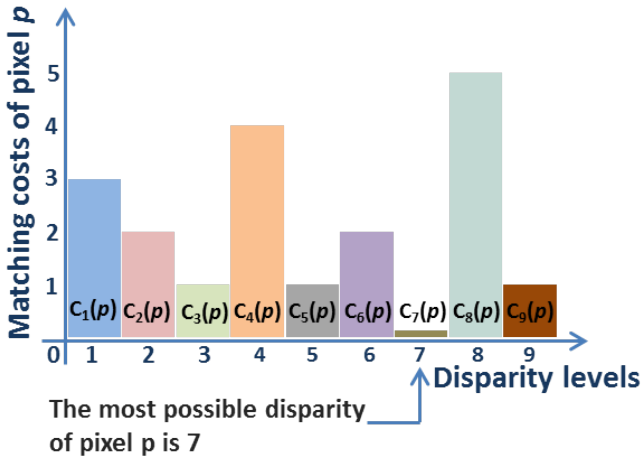


Figure 3.2: Winner-Takes-All.

3.1.2 Global algorithms

Different from local stereo matching algorithms which compute disparity independently for each pixel, global algorithms construct an energy function that contains the information of the whole image and iteratively optimize this function [10, 44, 45].

For a global algorithm, the process of computing the disparity map is treated as finding the solution of a pixel-labelling problem. The solution d can be obtained by minimizing a global energy function as shown in Equ. (3.1):

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d)\,, \tag{3.1}$$

where $E_{data}(d)$ and $E_{smooth}(d)$ denote the data term and smoothness term respectively; λ is the scale factor which controls the influence of smoothness term [10]. The data term $E_{data}(d)$ measures the similarity between pixels in the left and the right images

[30, 46]. A general expression for E_{data} is presented as follows:

$$E_{data}(d) = \sum_{x,y} C(x, y, d(x, y)) \,, \tag{3.2}$$

where $C(x, y, d(x, y))$ is the matching cost between the pixel located at (x, y) and its corresponding pixel at disparity $d(x, y)$. The smoothness term $E_{smooth}(d)$ indicates the smoothness assumption made by the algorithm. In order to model the smoothness assumption mathematically, only the disparity changes between neighbouring pixels are taken into account, as shown in Equ. (3.3):

$$E_{smooth}(d) = \sum_{x,y} \left[\rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))\right] \,, \tag{3.3}$$

where ρ denotes a monotonically increasing function of the differences in disparity. However, such smoothness term may lead to inappropriate smoothness at disparity boundaries and in occlusion areas. Intuitively, disparity boundaries tend to be also intensity edges. Hence, the smoothness term can be modified to take the intensity differences into account as well, as shown in Equ. (3.4):

$$E_{smooth}(d) = \sum_{x,y} \rho_d(d(x, y) - d(x + 1, y)) \cdot \rho_I(\|I(x, y) - I(x + 1, y)\|) \,, \tag{3.4}$$

where ρ_d denotes the monotonically increasing function of disparity differences; and ρ_I denotes the monotonically decreasing function of intensity differences.

Once the global energy function is defined, different algorithms can be employed to minimize the function. Since the stereo matching problem can be modelled as Markov Random Field (MRF) optimization problem or Conditional Random Field

(CRF) optimization problem, a variety of related algorithms have been proposed. Dynamic programming [25, 47], graph cuts [22, 23], belief propagation [27–29] are all popular global algorithms.

3.2 Non-local methods

Traditional local methods conduct cost aggregation by averaging the cost within a support region since they implicitly assume that pixels in the same support region have similar disparities. Due to the usage of local averaging techniques, window-based local algorithms suffer from "edge fatten" effect when the support regions cover the disparity boundaries, which is very similar to image filtering methods. Yoon and Kweon proposed an adaptive support weights technique to filter the cost volume and preserve the disparity boundaries effectively [42]. Their algorithm was reformulated as the term of a joint bilateral filtering method in [48]. Based on the aforementioned edge-aware filtering techniques, guided image filtering was proposed in [49] and further applied to solve image labelling problems [50, 51]. However, the support regions for those filtering-based local algorithms are still limited in a pre-defined fixed-size window.

Recently, a non-local stereo matching algorithm based on cost aggregation over MST was proposed by Yang [19, 20]. In this method, a pixel is able to receive proper weighted supports from other pixels on MST through a unique shortest path. Different from local algorithms, non-local methods have a better performance due to the cost aggregation over the whole image. Mei *et al.* conducted non-local aggregation over segment-tree (ST) instead of MST [52]. Cheng *et al.* proposed cross-trees with

priors to optimize the non-local cost aggregation procedure. These works proved that non-local algorithms outperform all the local methods in terms of matching accuracy.

Figure 3.3 shows the illustration of the non-local cost aggregation over a tree structure generated from a 5×5 image, where the center pixel (shown as the red circle) receives supports from other pixels of the image (represent in blue circles) through a unique path (shown as arrows).

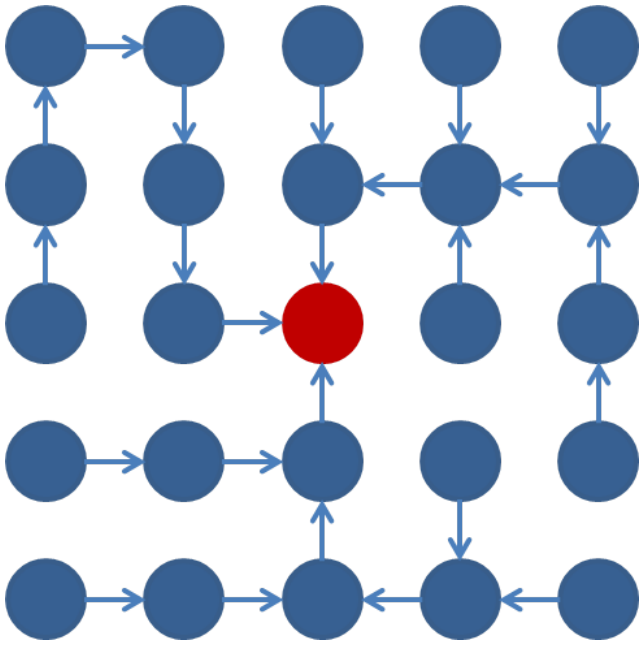


Figure 3.3: Illustration of non-local aggregation over tree structure.

3.2.1 Methods based on aggregation over MST

In Yang's non-local stereo matching framework, the reference image I is treated as a 4-connected, undirected graph $G = (V, E)$, where V and E represent the vertices and edges in the graph [19, 20]. Specifically, V are all the pixels in the reference

image and E are edges connecting neighbouring pixels. The edge weight w between neighbouring pixels s and r is computed by Equ. (3.5) [19] or Equ. (3.6) [20]:

$$w(s,r) = w(r,s) = |I_s - I_r|\,, \tag{3.5}$$

$$w(s,r) = w(r,s) = |I_s - I_r|^2\,, \tag{3.6}$$

where I_s and I_r represent the intensity values of s and r respectively.

Yang's method assumes that the intensity edges can be regarded as the depth edges. Therefore a minimum spanning tree (MST) can be generated from G by using Kruskal's algorithm to remove edges with large weights iteratively [53].

After the MST is constructed for image I, the similarity between two pixels can be defined using MST. The closer the two pixels are in this MST, the more similar they are in the image. The distance between two pixels p and q, denoted as $D(p,q) = D(q,p)$, is defined as the sum of edge weights along the shortest path between the two corresponding nodes on the MST. The similarity between pixel p and pixel q can then be defined by Equ. (3.7):

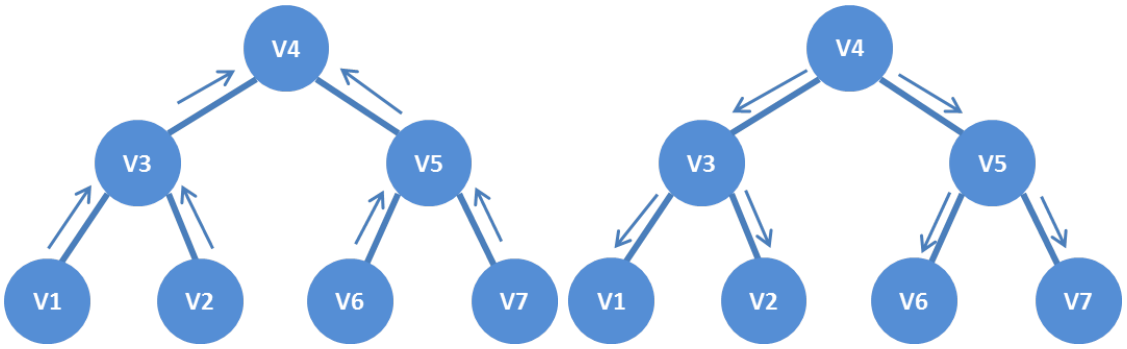
$$S(p,q) = S(q,p) = exp\left(-\frac{D(p,q)}{\sigma}\right)\,, \tag{3.7}$$

where σ is an constant used to adjust the similarity between p and q.

The final aggregated cost for pixel p over MST can be calculated by Equ. (3.8):

$$C_d^A(p) = \sum_{q \in I} S(p,q) \cdot C_d(q)\,. \tag{3.8}$$

Non-local cost aggregation on tree structure makes every pixel can receive proper weighted supports from all other pixels in image I, which outperforms window-based local algorithms. Using Equ. (3.8) to compute cost aggregation iteratively is feasible but time-consuming. Yang proposed a non-local cost aggregation method with linear time complexity by computing leaf-to-root and root-to-leaf cost aggregation successively [19, 20]. Figure 3.4 illustrates the two-step non-local aggregation on MST.



(a) Leaf-to-root aggregation (b) Root-to-leaf aggregation

Figure 3.4: Two-step non-local cost aggregation.

In leaf-to-root step (Figure 3.4 (a)), the intermediate aggregated cost $C_d^{A\uparrow}$ for pixel p is computed by Equ. (3.9):

$$C_d^{A\uparrow}(p) = C_d(p) + \sum_{Par(p_c)=p} S(p, p_c) \cdot C_d^{A\uparrow}(p_c)\,, \tag{3.9}$$

where $Par(p_c)$ denotes the parent node of p_c. After the first step, the root node (V_4 in Figure 3.4 (a)) receives proper weighted supports from all the other nodes while the rest of the nodes receive supports from their subtrees. Note that if node p has no child (leaf node), then $C_d^{A\uparrow}(p) = C_d(p)$.

In root-to-leaf step (Figure 3.4 (b)), the final aggregated cost C_d^A for pixel p is

computed by Equ. (3.10):

$$\begin{aligned} C_d^A(p) =& C_d^{A\uparrow}(p) + S(Par(p), p) \cdot [C_d^A(Par(p)) - S(Par(p), p) \cdot C_d^{A\uparrow}(p)] \\ =& S(Par(p), p) \cdot C_d^A(Par(p)) + [1 - S^2(Par(p), p)] \cdot C_d^{A\uparrow}(p)\,. \end{aligned} \tag{3.10}$$

If p is the root node of the MST, $C_d^A(p) = C_d^{A\uparrow}(p)$. Then the aggregated cost for each node can be obtained by Equ. (3.10) tracing from root node towards corresponding leaf nodes iteratively.

The computational complexity of this two-step aggregation algorithm is extremely low. For each pixel at each disparity level, only two addition/subtraction operations and three multiplication operations are required [19]. The computational complexity is $O(N \cdot L)$, where N is the number of pixels of the image and L is the number of disparity range.

After the non-local cost aggregation, the WTA strategy is conducted to obtain the disparity map.

A new disparity refinement algorithm was also proposed by Yang based on the non-local cost aggregation [19, 20].

After achieving the disparity maps for both left image and right image, a left-right cross check (also called consistency check) is conducted for each pixel to determine if it is stable or not. A stable pixel should have the same disparity value with the corresponding pixel in the other image. A new cost value is assigned for each pixel at each disparity level based on the result of cross check. The new cost for pixel p at

disparity level d is computed by Equ. (3.11):

$$C_d^{new}(p) = \begin{cases} |d - D(p)| & \text{if } p \text{ is stable and } D(p) > 0\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{3.11}$$

where D denotes the computed left disparity map.

The aforementioned non-local cost aggregation method is then employed to aggregate the new costs on the same MST. Since the new cost for all unstable pixels at each disparity level is 0, the disparity values of unstable pixels will only depend on the stable pixels.

3.2.2 Methods based on aggregation over Segment-Tree

Instead of employing MST, Mei *et al.* conducted non-local cost aggregation on a graph-based Segment-Tree (ST) [52]. Different from MST in which the graph connectivity is determined only with intensity differences between neighbouring pixels, ST further introduces a non-local rule: pixels in the same segment are more likely to share the same disparity value, thus the connectivity of pixels within the same segment should be enforced first [52]. ST can be constructed with three steps:

1. Divide the image pixels into several segments.

2. Build the subtree for each segment.

3. Link all subtrees to generate the final tree.

Step 1 is a typical segmentation problem which can be solved with any robust segmentation algorithms, such as normalized cuts [54, 55] and mean-shift clustering

[56, 57]. In Mei's algorithm, the graph-based segmentation method proposed by Felzenszwalb and Huttenlocher [35] is employed and further extended to handle the three-step ST construction. Step 2 enforces the connectivity within a segment while step 3 enforces the connectivity around each segment.

The reference image (left image) is treated as a 4-connected, undirected graph $G = (V, E)$. Similar to MST, V and E denote the pixels in the image and edges connecting neighbouring pixels respectively. For each edge $e \in E$, an associated weight w_e connecting pixels s and r is computed by Equ. (3.12):

$$w_e = w(s, r) = w(r, s) = |I_s - I_r| , \tag{3.12}$$

The required ST is defined as $T = (V, E')$, where E' is a subset of E. The construction of ST, which can also be regarded as selecting the proper subset E' from E, proceeds in three stages [52]:

- **Initialization**: According to the edge weights, the edges in E are sorted in a non-decreasing order. For each node v_i in V, a subtree T_i is created which only contains one node. At this stage, E' is an empty set.

- **Grouping**: At this stage, subtrees are merged into several bigger groups (segments) with a full scan of E. Suppose $e_j \in E$ is the edge connecting nodes v_p and v_q, the corresponding edge weight is w_{e_j}. If v_p and v_q belong to different subtrees T_p and T_q, and w_{e_j} satisfies the criterion proposed in [35], T_p and T_q will be merged into a new subtree $T_{p,q}$. Once two subtrees merge into a new subtree, the connecting edge e_j is included in E'. The aforementioned criterion

considers the similarity between two subtrees, as shown in Equ. (3.13):

$$w_{e_j} \leq min(Int(T_p) + \frac{k}{|T_p|}, Int(T_q) + \frac{k}{|T_q|}), \tag{3.13}$$

where $Int(T_p)$ and $Int(T_q)$ denote the maximum edge weight in T_p and T_q respectively; k denotes a constant parameter. After traversing all the edges in E, visually consistent segments are generated as the form of subtrees. At this time, the edges of the subtrees, which are already collected in E', are deleted from E.

- **Linking**: At this stage, each segment is linked by selecting more edges from E. All the edges left in E are scanned to search for the edges connecting different segments. If an edge connects two different subtrees, this edge will be collected in E' and these two subtrees will be merged together to generate a new subtree. When all the subtrees are finally merged into one tree which contains all the nodes in V, the search stops and ST $T = (V, E')$ is constructed. Note that since the edges in E are already sorted in a non-decreasing order, edges selected into E' to connect each subtrees are those with small edge weight.

It can be proved that each subtree is an MST of the corresponding segment after the grouping stage [35]. Since the edges selected in linking stage are those with small edge weights, the final ST is also proved to be an MST of the graph G [52].

After the ST is constructed, the non-local cost aggregation algorithm proposed by Yang [19, 20] is employed on ST. Then the commonly-used WTA strategy is employed to estimate the disparity map.

An enhanced ST was also proposed by Mei *et al.* which employed both color

information and initial disparity information [52].

The enhanced ST is based on the observation that neighbouring regions with different colors may still have similar disparities. In order to obtain robust non-local cost aggregation, those regions should be merged together. Using both color and disparity cues is also proved to be helpful for improving scene segmentation [58, 59].

Suppose the disparity map computed from non-local cost aggregation over aforementioned ST is denoted as D, all the edge weights are then updated using Equ. (3.14) [52]:

$$w_e = \lambda \frac{|I(s) - I(r)|}{\Delta_I} + (1 - \lambda) \frac{|D(s) - D(r)|}{\Delta_D}, \tag{3.14}$$

where w_e denotes the edge weight of edge e; e is the edge connecting pixel s and pixel r; Δ_I and Δ_D represent two constant normalization parameters; $\lambda \in [0, 1]$ represents the scale parameter for controlling the contribution of color information.

By employing the same ST construction algorithm on the updated graph, an enhanced ST can be generated. Conducting non-local cost aggregation on this enhanced ST, an improved disparity map can be achieved after WTA strategy.

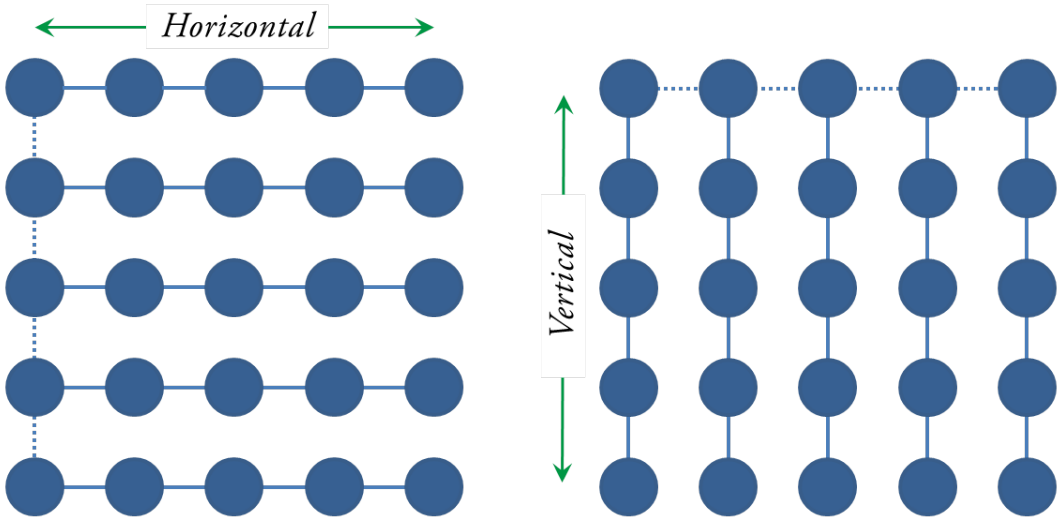
3.2.3 Methods based on aggregation over cross-trees

Recently, Cheng *et al.* proposed a novel cross-trees structure to conduct non-local cost aggregation algorithm [60, 61]. The cross-trees structure consists of two unique crossed trees which are independent of any local or global property of the image: a horizontal tree and a vertical tree. For convenience, *cross-trees* is used to denote the two crossed trees in the rest of this thesis.

An explicit smoothness assumption is made by employing truncated edge weights during the construction of cross-trees, which may lead to "edge fatten" effect if directly performing non-local cost aggregation on such tree. In order to restrict the false cost aggregation across disparity boundaries, edge prior and superpixel prior are also proposed.

Based on different priors being used, Cheng's method contains two algorithms: Cross-E and Cross-SP, which denote cross-trees with edge prior and cross-trees with superpixel prior respectively.

The reference image I is treated as a 4-connected, undirected graph $G = (V, E)$, where V and E denote all the pixels in I and all the edges connecting neighbouring pixels respectively [19, 20].



(a) Horizontal tree (b) Vertical tree

Figure 3.5: Illustration of cross-trees structure.

Figure 3.5 illustrates cross-trees structure of a 5×5 graph. Figure 3.5 (a) is the horizontal tree and Figure 3.5 (b) is the vertical tree. Blue circles represent the

nodes and the lines connecting pixels are the edges of the tree structure. Note that the dash lines are extra edges used to connect different rows or columns so that the tree structure can be constructed. In order to prevent the cost aggregation between different rows or columns, the weights of the edges marked with dash lines are set to be a very large number in practice.

Figure 3.6 shows the illustration of performing non-local cost aggregation strategy on cross-trees structure. By performing non-local cost aggregation successively on horizontal tree and vertical tree, a pixel (represented with red circle) is able to receive weighted supports from other pixels of the image (represented with blue circles) through a unique path (shown as arrows). The green arrows represent the cost aggregation on horizontal tree and the orange arrows represent the cost aggregation on the vertical tree.

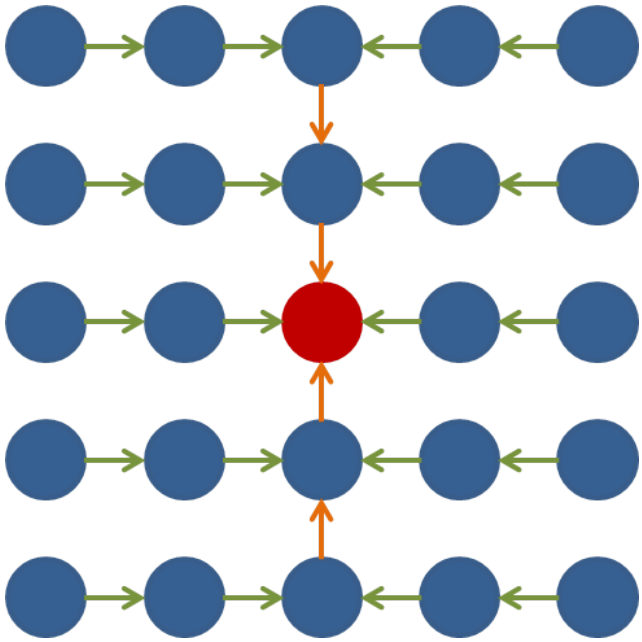


Figure 3.6: Illustration of non-local aggregation on cross-trees structure.

Instead of generating a tree structure only based on local pixel similarity, cross-trees structure aims to be independent of the image while maintains the spatial smoothness of the disparity map. Each edge is assigned a truncated weight as shown in Equ. (3.15), which corresponds to the global smoothness assumption:

$$w(s,r) = w(r,s) = min(|I_s - I_r|, \tau)\,, \tag{3.15}$$

where $w(s,r)$ is the weight of the edge connecting neighbouring pixels s and r; I_s and I_r are the intensity values of pixel s and pixel r respectively; τ is the truncation threshold of the intensity differences of neighbouring pixels.

However, assuming disparity smoothness everywhere in the image may cause false cost aggregation across disparity boundaries. Hence, two different edge priors, edge prior and superpixel prior, are employed to locate the potential disparity boundaries [60, 61].

(a) Edge prior (b) Superpixel prior

Figure 3.7: Different priors incorporated into non-local framework.

Edge prior is a common-used prior as shown in Figure 3.7 (a), where the edges

(marked with white lines) are detected by Canny edge detector. Since Canny detector is sensitive to intensity changes, many false edges in highly textured regions are also detected. Assuming all the edges detected by edge detector to be potential disparity boundaries will degrade the performance of non-local cost aggregation in highly textured regions.

Figure 3.7 (b) shows the superpixel prior obtained by using SLIC algorithm [62]. Since superpixels are compact and regular, disparity boundaries are fully connected. Meanwhile, many false edges in highly textured regions can also be removed.

Hence, the edge weight function (Equ. (3.15)) can be rewritten by Equ. (3.16):

$$w(s,r) = w(r,s) = \begin{cases} |I_s - I_r| & \text{if } e(s,r) \cap \text{ the prior}\,, \\ min(|I_s - I_r|, \tau) & \text{otherwise}\,, \end{cases} \tag{3.16}$$

where $e(s,r) \cap$ the prior denotes that the edge connecting pixel s and pixel r crosses the prior; τ is the truncation threshold of intensity differences if the edge connecting pixels r and s does not cross the prior.

By employing the global smoothness assumption along with proper prior, costs can be aggregated within planar surfaces and the disparity boundaries can also be preserved [60, 61].

Figure 3.8 illustrates how to perform the non-local cost aggregation for pixel p (represented with red circle) on cross-trees structure with a prior. In this example, the curve represents the prior; the green arrows represent the cost aggregation flow on horizontal tree and the orange arrows represent the cost aggregation flow on vertical tree. When the cost of pixel q_2 is aggregating to pixel p through path $P(q_2, p)$, the

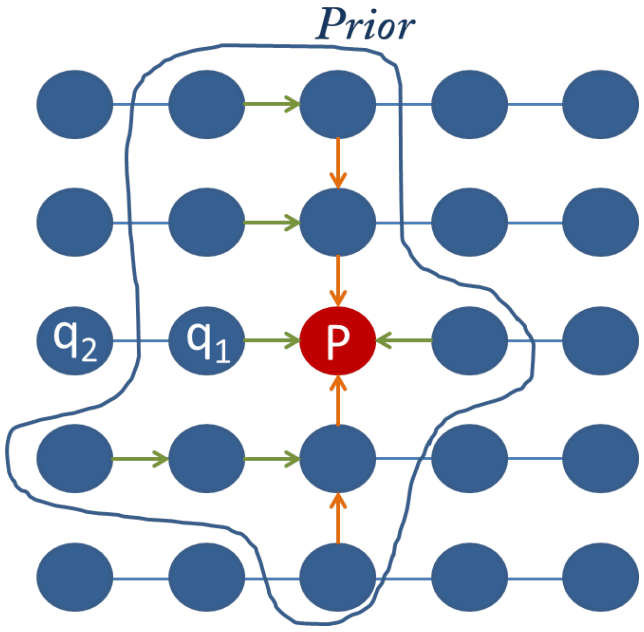


Figure 3.8: Illustration of non-local aggregation on cross-trees structure with a prior.

edge $e(q_2, q_1)$ on this path crosses the prior, thus the edge weight is the intensity difference of pixels q_2 and q_1. If the intensity difference between q_2 and q_1 is large, the distance between pixel q_2 and center pixel p is also large. Thus the contribution of pixel q_2 and contributions of pixels even farther to center pixel p will be much less. For pixel p, the region within the prior is its support region. Since the prior can be of any shape, the sizes and shapes of support regions can be arbitrary, which is very difficult for traditional local window-based stereo matching algorithms.

Note that for superpixel prior, the cost aggregation in large non-texture regions will not be terminated since the intensity differences in such regions are already very low.

After performing the non-local cost aggregation on cross-trees structure with a prior, the disparity map can be obtained by the WTA strategy.

3.3 Summary

In recent decades, a large variety of algorithms have been proposed for solving dense stereo matching problem. These algorithms can be broadly divide into two classes: local algorithms and global algorithms. Global algorithms generally produce more accurate disparity maps than local algorithms. However, they are much slower than local algorithms. More recently, non-local algorithms are proposed to balance the accuracy and efficiency. Several state-of-the-art non-local algorithms were presented in this chapter. In the rest of this thesis, we will propose a novel non-local stereo matching method that outperforms the current state-of-the-art non-local algorithms.

Chapter 4

Proposed algorithm

This chapter describes the principles and implementation of our proposed stereo matching method, which is mainly based on the non-local cost aggregation over an edge-aware truncated minimum spanning tree (T-MST). The flowchart of this method is shown in Figure 4.1.

The proposed method consists of three steps to generate the final disparity map:

1. Matching cost computation: Combine the truncated absolute differences (TAD) and the histogram of oriented gradient (HOG) to formulate an efficient and robust cost function;

2. Non-local cost aggregation: Perform non-local cost aggregation over edge-aware T-MST which is generated based on a novel hybrid prior;

3. Adaptive disparity refinement: Conduct adaptive non-local refinement based on the pixel stability.

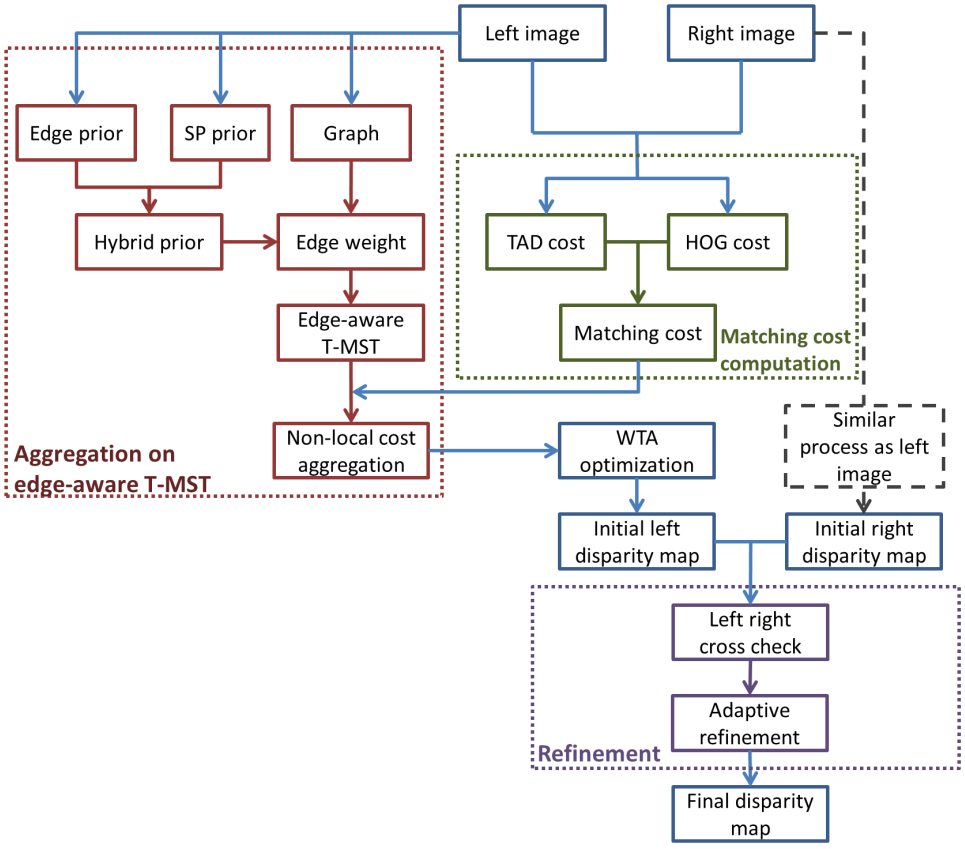


Figure 4.1: The flowchart of the proposed stereo matching method.

In this chapter, we use the stereo images from Middlebury benchmark [10] as our test data sets, which are "Tsukuba" (288×384), "Teddy" (375×450), "Venus" (383×434) and "Cones" (375×450) [10, 63].

4.1 Pixel cost computation

The matching cost for each pixel at each disparity level between the left image I^l and the right image I^r is computed. The matching cost function proposed in this thesis is a convex combination of the truncated absolute differences (TAD) [50, 51] and the norm of the differences between the vectors of histogram of oriented gradient (HOG) [64, 65].

The pixel matching cost function is shown in Equ. (4.1):

$$C_d(p) = \gamma C_d^{TAD}(p) + (1-\gamma) C_d^{HOG}(p)\,, \tag{4.1}$$

where $C_d(p)$ represents the matching cost of pixel p at disparity level d; $C_d^{TAD}(p)$ and $C_d^{HOG}(p)$ represent TAD cost term and HOG cost term respectively, which will be discussed in detail in the following subsections; γ is a scale factor used to control the contribution of TAD cost term.

4.1.1 TAD cost computation

Compared with traditional matching cost function AD, TAD is more robust to random image noise. In our proposed method, we employ the TAD of both intensity and gradient, which has shown to be robust to outlier pixels and illumination variation

[50, 51, 66–68], to be a part of the proposed matching cost function.

Similar to [69], the TAD matching cost of a pixel p and its corresponding pixel p_d at disparity level d is defined as a convex combination of the intensity differences e_i and the gradient dissimilarity e_g, as shown in Equ. (4.2) [66]:

$$C_d^{TAD}(p) = \beta \cdot e_i(p, p_d) + (1 - \beta) \cdot e_g(p, p_d)\,, \tag{4.2}$$

where e_i and e_g are given as follows:

$$e_i(p, p_d) = min(|I^l(p) - I^r(p_d)|, T_i)\,, \tag{4.3}$$

$$e_g(p, p_d) = min(|I_g^l(p) - I_g^r(p_d)|, T_g)\,, \tag{4.4}$$

where $I^l(p)$ and $I^r(p_d)$ denote the intensity values of the corresponding pixels; $I_g^l(p)$ and $I_g^r(p_d)$ denote the horizontal gradients of the corresponding pixels; T_i and T_g are two empirical truncation parameters for intensity and gradient respectively; β is a weight factor which balances the color and gradient terms. In all the experiments of this thesis, we follow the parameter setting in [50, 51], where β is set to 0.11; T_i and T_g are set to 7 and 2 respectively.

4.1.2 HOG cost computation

HOG was firstly proposed to accurately describe object feature for image recognition and object detection [64], thus it can also be used for computing matching cost [65]. An improved HOG, which is efficient and robust to linear radiometric variation, was

proposed in [65] for stereo matching cost measurement.

Suppose the radiation distortion is linear within a very small window, for instance, a 3×3 window. The linear radiation distortion can be defined by Equ. (4.5) [65]:

$$I^l(p) = c \cdot I^r(p_d) + t\,, \tag{4.5}$$

where p and p_d are corresponding pixels in the left and the right images at disparity level d; $I^l(p)$ and $I^r(p_d)$ represent the intensity values of pixel p in the left image and pixel p_d in the right image respectively; c and t are the scale factor and the translation factor of the linear radiation distortion model respectively.

By computing the gradient within a small window, the translation factor t can be removed. In order to further eliminate the scale factor c, the Sobel operator is used to calculate the gradient direction. Hence, Equ. (4.5) can be rewritten by Equ. (4.6) [65]:

$$\theta^l(p) = \theta^r(p_d)\,, \tag{4.6}$$

where $\theta^l(p)$ and $\theta^r(p_d)$ represent the gradient directions of pixel p and pixel p_d respectively.

The gradient direction, as a linear radiometric invariant metric, is defined by Equ. (4.7):

$$\theta(q) = arctan\left(\frac{G_y(q)}{G_x(q)}\right)\,, \tag{4.7}$$

where $\theta(q)$ denotes the gradient direction of pixel q; $G_y(q)$ and $G_x(q)$ represent the

vertical and horizontal gradient of q respectively. The range of gradient direction is $[0, 360°)$.

For a pixel p, the basic description cell is defined as a $W \times W$ window centred at it. The gradient directions for all pixels within the window are computed. We divide the gradient direction range into 12 bins with a step size of 30°. Then a gradient direction histogram is computed based on the counts of pixels in each bin.

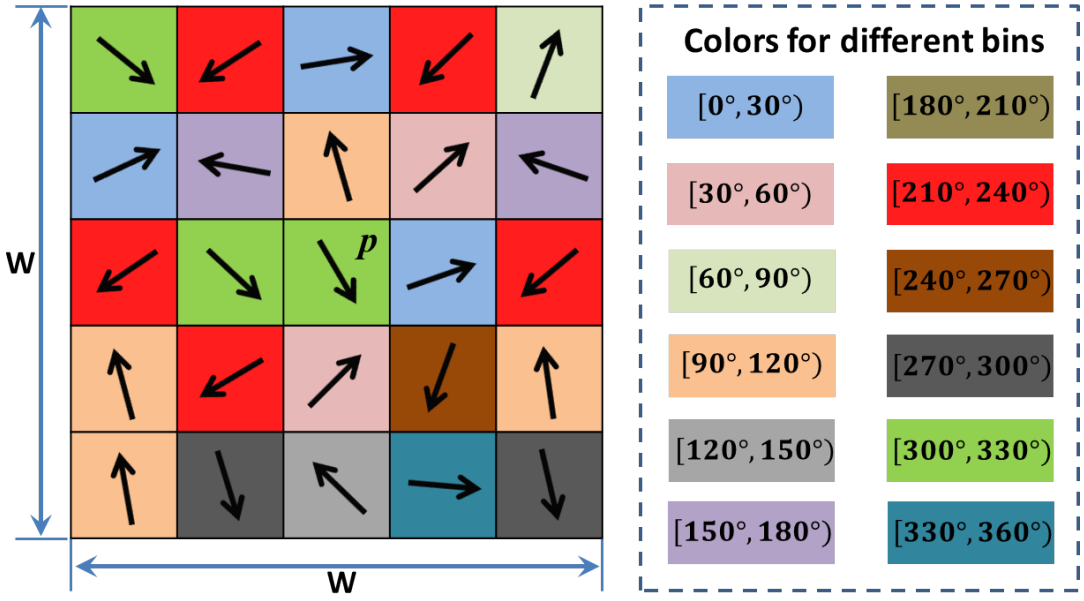


Figure 4.2: Gradient direction.

Given a 5×5 window centred at pixel p, Figure 4.2 shows an example of computing the gradient directions for all the pixels within this description cell. Each rectangle represents a pixel. The angle of each arrow indicates the gradient direction of the corresponding pixel. Different background colors of the rectangles indicate different bins.

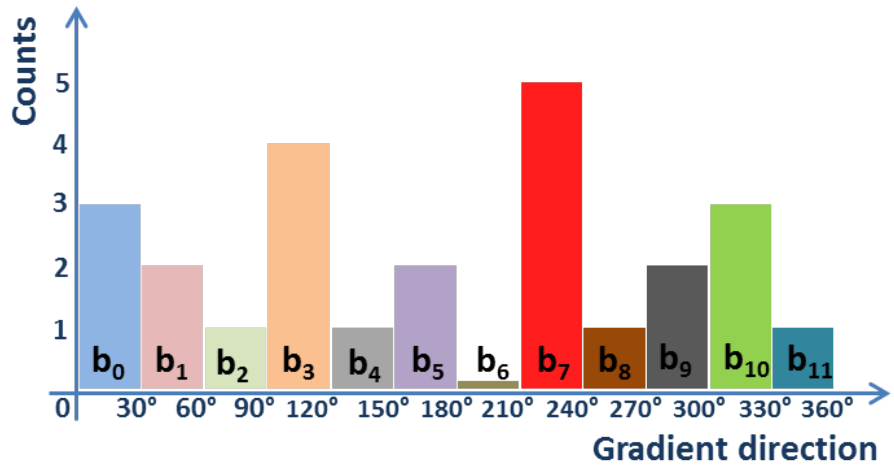


Figure 4.3: Histogram of the gradient directions.

After the gradient directions of all the pixels in the description cell are computed, the histogram of gradient directions can be generated. Figure 4.3 represents the histogram generated from Figure 4.2. HOG feature descriptor of pixel p can then be constructed with a vector as shown in Equ. (4.8):

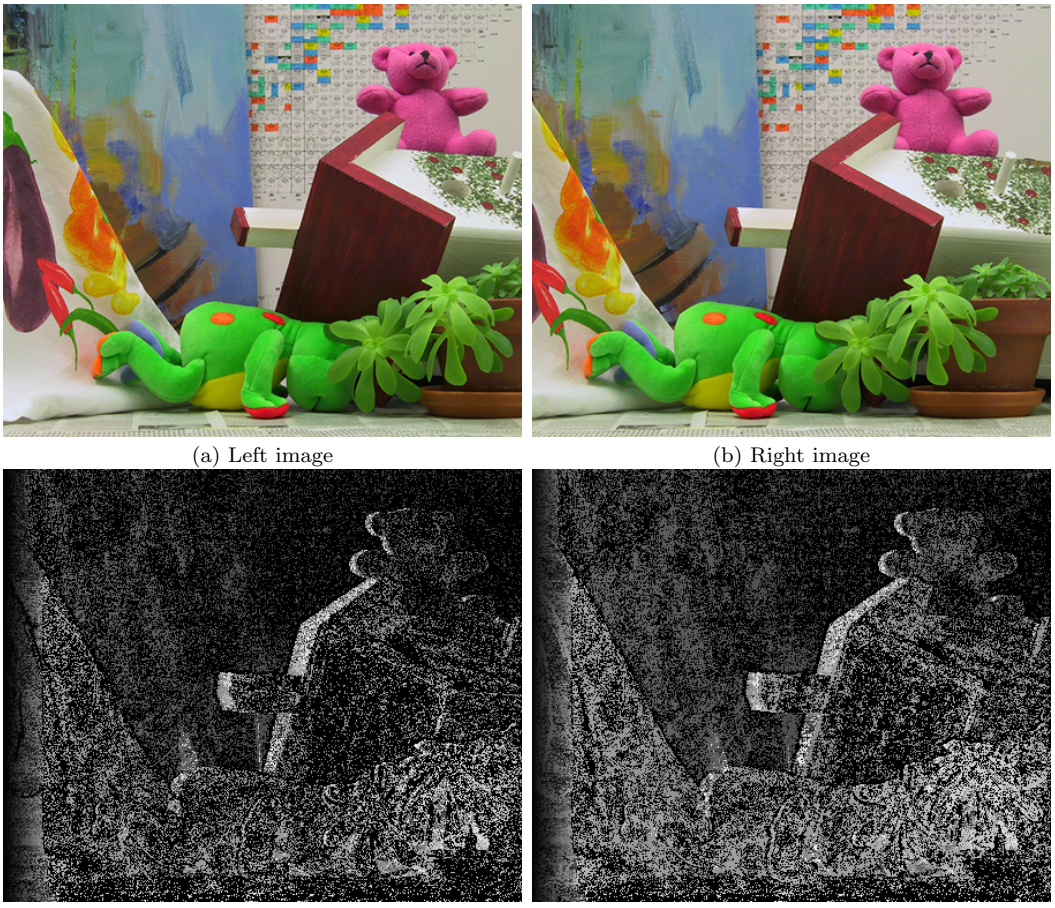
$$V_{HOG}(p) = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}) , \tag{4.8}$$

where $V_{HOG}(p)$ denotes the HOG feature descriptor of pixel p; $b_i (i = 0, 1, 2, \ldots, 11)$ represents the corresponding value in gradient direction histogram.

The HOG matching cost function is then defined as the distance between the HOG feature descriptors of the corresponding pixels, as shown in Equ. (4.9):

$$C_d^{HOG}(p) = \left\| V_{HOG}^l(p) - V_{HOG}^r(p_d) \right\| , \tag{4.9}$$

where $V_{HOG}^l(p)$ and $V_{HOG}^r(p_d)$ are HOG feature vectors of corresponding pixels.



(a) Left image (b) Right image

(c) Raw disparity map with TAD cost (a) Raw disparity map with proposed matching cost

Figure 4.4: Raw disparity maps for Teddy with different matching cost ($\gamma = 0.4$, window size $= 5$, searching range $= 53$).

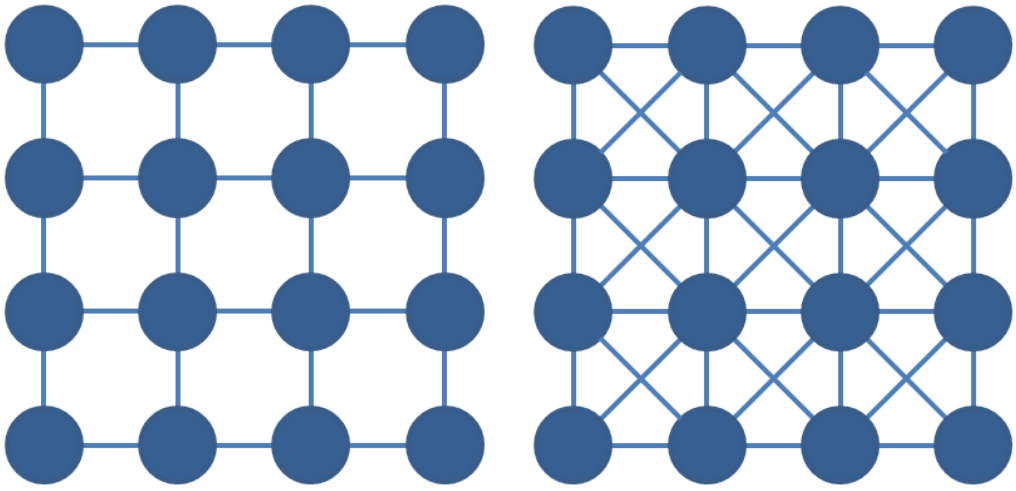
Figure 4.4 shows raw disparity maps for data set "Teddy" obtained from different matching cost computation methods. Raw disparity map is obtained by employing WTA strategy on computed matching costs. Figure 4.4 (a) and (b) are left image and right image respectively. Figure 4.4 (c) is the raw disparity map computed with TAD matching cost function. Figure 4.4 (d) is the raw disparity map computed with our proposed matching cost function. Black regions in raw disparity maps indicate inaccurate pixels with error threshold 1 in non-occluded regions (compared with ground truth disparity maps). The percentages of inaccurate pixels of Figure 4.4 (c) and (d) are 73.40 and 63.74 respectively, which shows that the HOG cost term improves the accuracy of computed matching costs. Note that all the following steps (such as aggregation and post-processing) in stereo matching algorithms are actually performing refinement on computed matching costs. Hence an accurate and robust matching cost function will benefit the final disparity map.

4.2 Cost aggregation on edge-aware T-MST

Our work basically follows Yang's non-local framework [20]. A significant difference between our algorithm and Yangs's algorithm is that we employ a different tree structure: edge-aware T-MST. Motivated by the recent non-local cost aggregation method proposed by Cheng *et al.* [60, 61], explicit global smoothness assumption is made in our algorithm by employing truncated edge weight between neighbouring pixels. Since assuming smoothness everywhere may cause false cost aggregation across disparity boundaries, a novel hybrid edge prior which combines edge prior and superpixel prior is proposed to preserve the potential disparity boundaries.

In our proposed stereo matching framework, the reference image I is treated as a eight-connected, undirected graph $G = (V, E)$, where V and E represent the vertices and edges in the graph. Specifically, V are all the pixels in the reference image and E are edges connecting neighbouring pixels. The edge weights assigned to edges in E are defined according to pixel intensity similarities and the proposed hybrid edge prior.

Note that, the eight-connected graph, which is different from the four-connected graph where pixels are only connected horizontally and vertically, ensures pixels being connected horizontally, vertically and diagonally. Performing non-local cost aggregation over tree structure generated from an eight-connected graph allows message passing through more directions. Figure 4.5 shows the examples of different graph structures, where blue circles represent the pixels and lines represent the connecting edges between neighbouring pixels.



(a) Four-connected graph (b) Eight-connected graph

Figure 4.5: Different graph structures.

The basic prototype for calculating the edge weight w between neighbouring pixels s and r is shown in Equ. (4.10) [20]:

$$w(s,r) = w(r,s) = |I_s - I_r|^2 , \tag{4.10}$$

where I_s and I_r represent the intensity value of s and r respectively. This function assumes that intensity edges can be regarded as depth edges. However, in highly textured regions, despite of large intensity difference, disparity maps are spatially smooth in most cases.

(a) Reference image (b) Ground truth

Figure 4.6: Highly textured regions with smooth disparity changes.

Figure 4.6 shows examples of textured regions with smooth disparity changes (marked with red and yellow rectangles). Figure 4.6 (a) is the original left image of data set "Teddy"; Figure 4.6 (b) is the corresponding ground truth disparity map.

In order to be consistent with the fact that disparities are mostly spatially smooth, explicit smoothness assumption is made in our proposed algorithm. Similar to traditional global methods, the explicit smoothness assumption can be made by assigning

trucated weight to each edge as shown in Equ. (4.11):

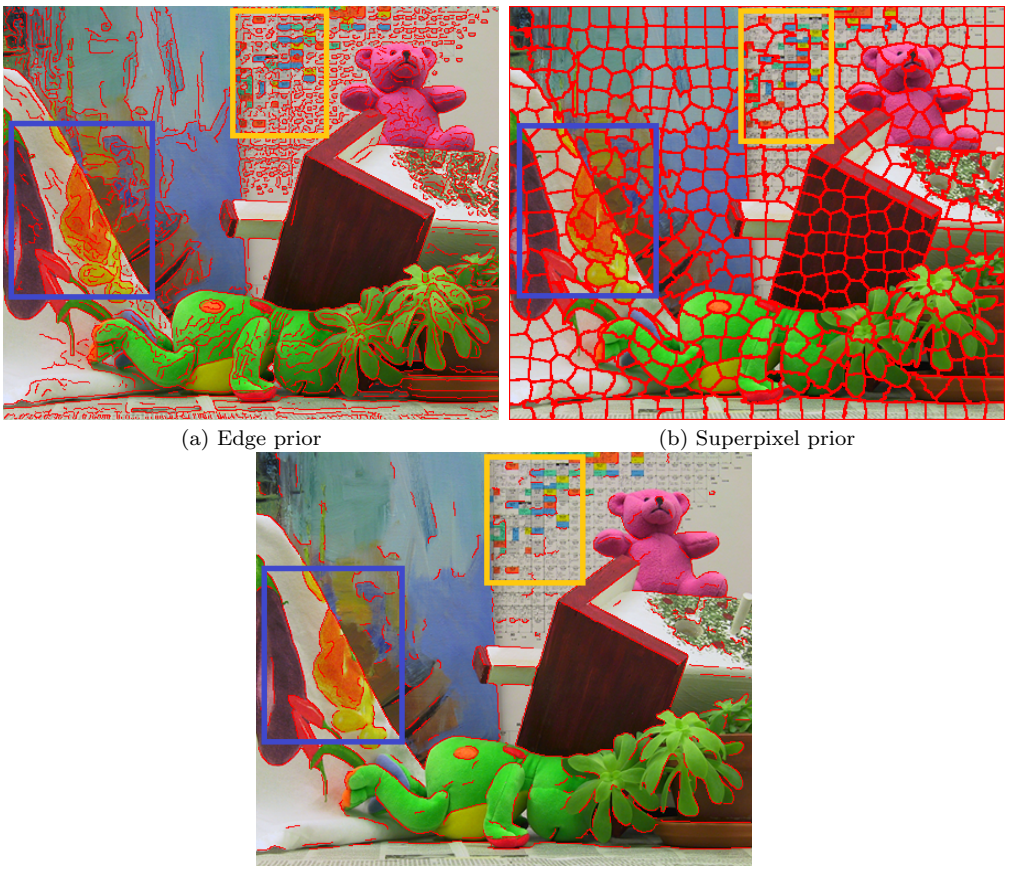
$$w(s,r) = w(r,s) = min(|I_s - I_r|^2, \tau)\,, \tag{4.11}$$

where τ is a truncation threshold of two neighbouring pixels s and r. However, the aggregation over a tree structure generated from truncated edge weight will suffer from "edge fatten" effect since it assumes disparity smoothness at every point. Hence proper prior is needed to indicate potential disparity boundaries.

A commonly used prior is the edge prior as shown in Figure 4.7 (a). Common edge detector, such as Canny edge detector, is sensitive to intensity changes so that many false edges are also detected. If all the color edges are considered to be disparity edges, the cost aggregation in highly textured regions will be degraded [60, 61].

Another recent popular prior is superpixel prior proposed by Cheng *et al.* [60, 61]. Since superpixels are regular and compact even in highly textured regions, many false edges can be avoided. Meanwhile, the disparity boundaries are fully connected in superpixel prior. Figure 4.7 (b) shows the superpixel prior computed for Teddy with superpixel size = 300.

In order to make full use of the advantages of both edge proir and superpixel prior, a novel hybrid edge prior is proposed. We assume that only edges detected by both edge detector and superpixel are considered to be disparity boundaries, as shown in Figure 4.7 (c). This hybrid edge is able to remove most false edges in textured regions and keep true depth boundaries to a great extend. Examples can be seen in Figure 4.7 marked with yellow and blue rectangles. Hence, Equ. (4.11) can be rewritten by



(a) Edge prior (b) Superpixel prior

(c) Hybrid edge prior

Figure 4.7: Different priors for non-local framework.

Equ. (4.12):

$$w(s,r) = \begin{cases} |I_s - I_r|^2 & \text{if } e(s,r) \cap \text{ the prior}\,, \\ min(|I_s - I_r|^2, \tau) & \text{otherwise}\,, \end{cases} \tag{4.12}$$

where $e(s,r) \cap$ *the prior* means the edge between s and r crosses the hybrid edge prior; τ is a truncation threshold.

After assigning edge weight to each edge in E with Equ. (4.12), the proposed edge-aware T-MST can be generated by applying Kruskal's algorithm [53].

Once the edge-aware T-MST is constructed, computed matching costs can be aggregated for each pixel. Non-local cost aggregation scheme as presented in Chapter 3 is then performed on edge-aware T-MST.

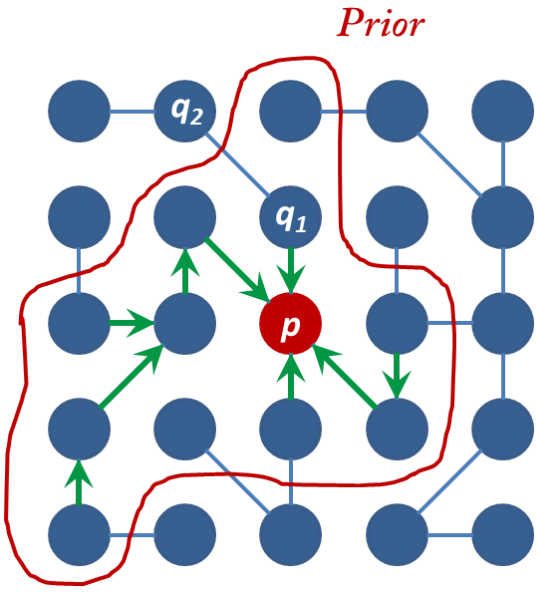


Figure 4.8: Non-local cost aggregation over edge-aware T-MST.

Figure 4.8 shows an illustration of non-local cost aggregation for pixel p (the red circle) on the edge-aware T-MST. Blue circles and lines represent pixels and connecting edges in the tree structure respectively. Green arrows represent the cost aggregation flow from other pixels to pixel p. During the non-local cost aggregation, the prior (represented with the red curve line) selects optimal support region for pixel p. The weights of the edges that cross the prior will be raised sharply if the intensity differences are large. For instance, the distance between q_2 and p is much larger than the distance between q_1 and p if the color dissimilarity between q_1 and q_2 is large, which suppresses the cost aggregation across the proposed prior.

Finally, the WTA strategy is employed to obtain the initial disparity map.

4.3 Adaptive refinement

After obtaining the initial disparity map for both left image and right image, a consistency check (also called cross check or left right check) is used to divide all the pixels into stable or unstable pixels. A stable pixel means that the corresponding pixel in the other image has the exact same disparity value. In order to achieve a better performance, all the unstable pixels should be interpolated in a proper way.

An adaptive non-local refinement scheme is proposed in this thesis to make full use of the stabilities of all the pixels. Let D denote the initial disparity map of left image, a new truncated cost is computed for each pixel [65]:

$$C_d^{new}(p) = \begin{cases} min(|d - D(p)|, \tau_{cc}) & \text{if } p \text{ is stable}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.13}$$

where $C_d^{new}(p)$ denotes the new cost value for pixel p at disparity level d; τ_{cc} represents the truncated threshold.

Since the new matching costs for unstable pixels at all disparity levels are set to be 0, the disparity of unstable pixels will completely depend on stable pixels.

Based on Yang's non-local refinement, which re-implements the cost aggregation method described in Chapter 3, we further update the edge-aware T-MST to a directed tree structure. In order to impose restrictions on cost aggregation based on pixel stabilities, we redefine the similarity calculation function between neighbouring pixels.

The distance $D(p,q) = D(q,p)$ between pixel p and pixel q is defined as the sum of edge weights along the path connecting p and q. When the cost is aggregated from pixel p to its neighbouring pixel q, similarity between p and q is redefined by Equ. (4.14):

$$S_n(p,q) = \begin{cases} \varphi \cdot exp\left(-\frac{D(p,q)}{\sigma}\right) & \text{if } p \text{ is unstable and } q \text{ is stable}, \\ exp\left(-\frac{D(p,q)}{\sigma}\right) & \text{otherwise}, \end{cases} \tag{4.14}$$

where $S_n(p,q)$ is the updated similarity between p and q; σ is a constant used to adjust the similarity between p and q; φ is a constant with range [0,1) used for suppressing the cost aggregation from unstable pixel to stable pixel.

Figure 4.9 illustrates the adaptive non-local refinement. In Figure 4.9, the green circles and red circles indicate stable pixels and unstable pixels respectively; the lines connecting pixels represent the aggregation paths; the arrows represent the directions of aggregation; green arrows represent higher aggregation weight compared with red

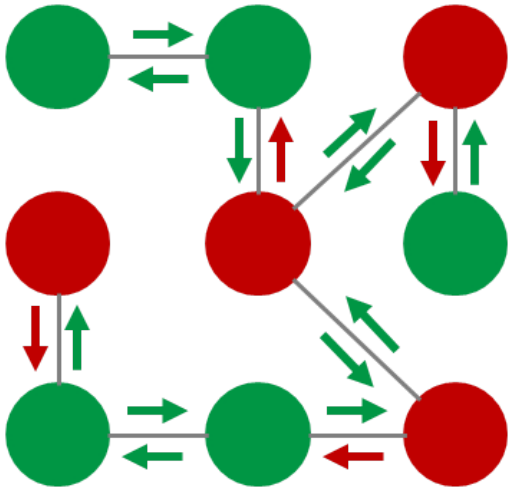


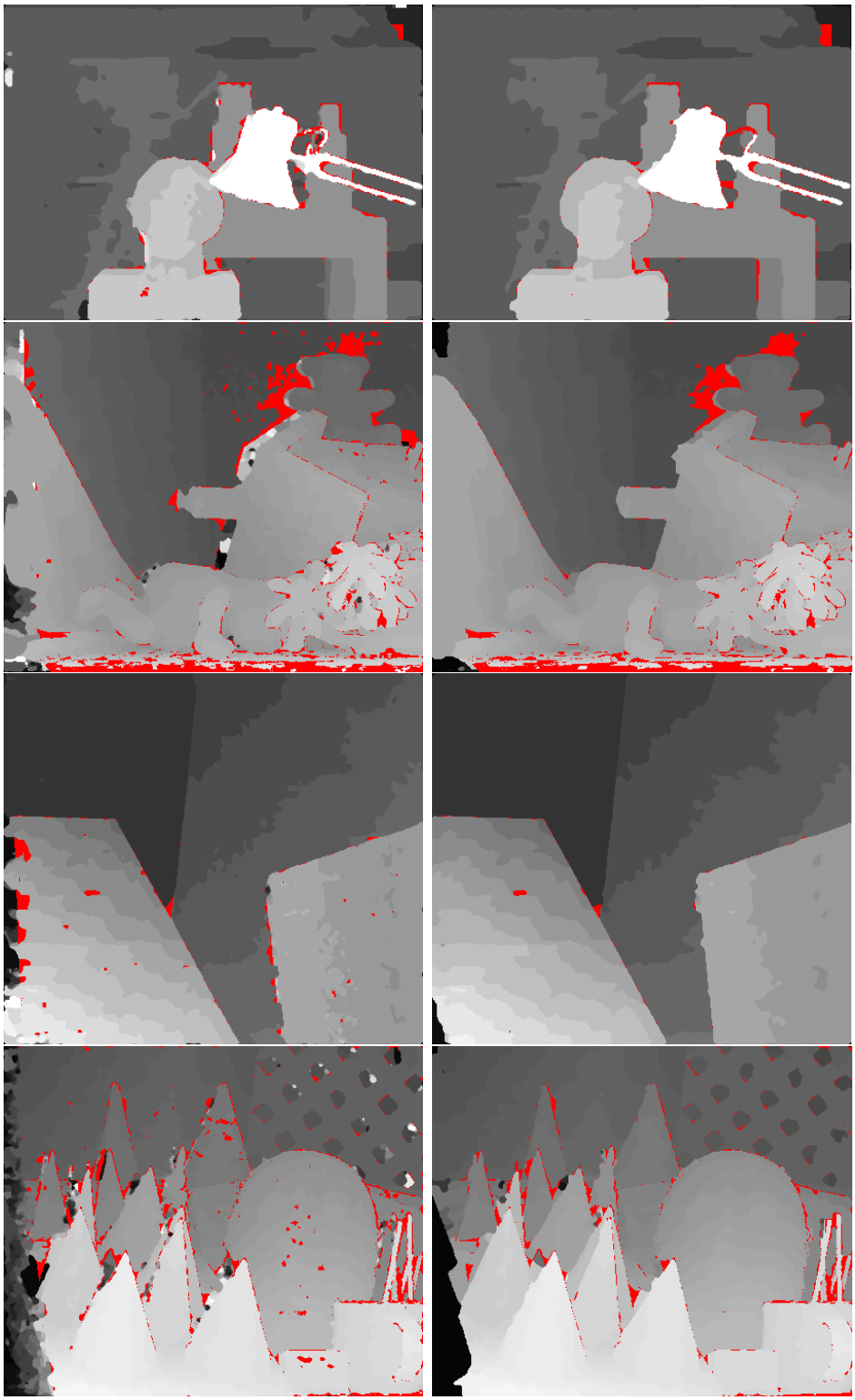
Figure 4.9: Adaptive non-local refinement.

arrows.

After the adaptive non-local refinement, the final disparity map is generated by implementing the WTA strategy again.

Figure 4.10 shows the performance of our proposed adaptive refinement scheme on standard Middlebury data sets [10, 63]. Starting from the first row, the test image sets are: "*Tsukuba*", *"Teddy"*, *"Venus"* and *"Cones"*. The images in the left column are initial disparity maps generated without refinement. The images in the right column are final disparity maps after adaptive refinement. Red regions represent the inaccurate pixesl with error threshold 1 in non-occluded regions. It can be seen that a noticeable portion of inaccurate pixels have been refined after the adaptive refinement.

Additionally, relative quantitative evaluations of proposed algorithm with or without refinement on standard Middlebury data sets are also presented in Table 4.1. For



(a) Proposed method **without** adaptive refinement

(b) Proposed method **with** adaptive refinemen

Figure 4.10: Performance of the proposed refinement.

each data set, the percentage of inaccurate pixels with threshold 1 in three different kinds of regions ("non-occ", "all" and "disc" denote non-occluded regions, all regions and discontinuous regions respectively) are calculated. The average errors on different data sets are presents in the last column.

As can be seen from Table 4.1, the proposed adaptive refinement scheme improves the performance of our stereo matching algorithm in terms of accuracy in all kinds of regions being evaluated.

Table 4.1: Quantitative evaluation of the proposed algorithm (with or without refinement) on standard Middlebury data set.

Method	Tsubuka			Teddy			Venus			Cones			Avg-error
	non-occ	all	disc	non-occ	all	disc	non-occ	all	disc	non-occ	all	disc	
No refine	1.61	2.93	7.93	6.52	12.84	15.36	0.97	1.93	7.26	4.14	13.48	11.13	7.18
Refine	**1.59**	**2.18**	**7.25**	**5.42**	**10.47**	**12.74**	**0.18**	**0.48**	**1.87**	**2.61**	**9.25**	**7.78**	**5.15**

4.4 Summary

In this chapter, our proposed non-local stereo matching algorithm is presented in three steps: matching cost computation, non-local cost aggregation on edge-aware T-MST and adaptive non-local refinement.

For matching cost computation, a novel cost function which combines TAD term and HOG term is proposed in this chapter. This new matching cost function is efficient and robust to outlier pixels and illumination variation. A novel tree structure, edge-aware T-MST, is also proposed for performing non-local cost aggregation. The proposed edge-aware T-MST promotes the aggregation in highly textured regions and in large textureless planar regions by employing truncation threshold. Meanwhile, the

"edge fatten" effect is suppressed by employing a novel hybrid edge prior to indicate potential disparity boundaries. Additionally, an adaptive non-local refinement scheme is proposed to make full use of pixel stabilities for more accurate final disparity maps.

Chapter 5

Experimental results

In this chapter, we evaluate and compare the performance of our proposed algorithm (denoted as T-MST) and other five non-local algorithms, which are: Yang's aggregation on MST (denoted as MST) [19, 20]; aggregation on ST (including two methods, denoted as ST1 and ST2) [52]; aggregation on cross-trees with priors (including two methods, denoted as Cross-E and Cross-SP) [60, 61].

All the algorithms are performed and tested on Middlebury data sets [10], details of the data sets being used in this thesis are presented in Section 5.1. The parameters used in the proposed algorithm are presented in Section 5.2. For other non-local stereo matching algorithms, the parameters are the same as they are in the corresponding papers. The quantitative evaluations and the visual qualities of the experimental results are presented in Section 5.3.

5.1 Data set

In our experiments, we employ 30 Middlebury data sets [10, 63, 70–72], including common-used standard Middlebury data sets (*Tsukuba, Venus, Teddy and Cones*), to reliably evaluate the performance of our proposed method and five non-local stereo matching methods in terms of accuracy.

Figure 5.1 presents the Middlebury data sets used in this thesis. Only the reference image (the left image) of each data set is presented due to space limitations. Note that, all the images listed in Figure 5.1 are rescaled to the same size for aesthetic considerations. The actual size of each data set is presented under the image along with the image name.

Figure 5.2 presents the ground truth disparity maps of the corresponding Middlebury data sets in the same order as in Figure 5.1. The images are also rescaled to the same size for better visual experience. The accurate ground truth disparity maps can be obtained with sensors or structured light [63]. Note that the black regions are uncertain pixels which are not taken into account in accuracy evaluation.

These 30 Middlebury data sets include different challenging structures, such as large textureless regions (*Bowling1, Bowling2, Midd1, Midd2, Plastic*), repeated patterns (*Aloe, Cloth1, Cloth2, Cloth4*), small occluded regions (*Art, Cones, Laundry*). Employing these data sets to test and compare the performance of stereo matching algorithms can give us comprehensive and reliable evaluation results.

Figure 5.1: Reference images of 30 Middlebury data sets.

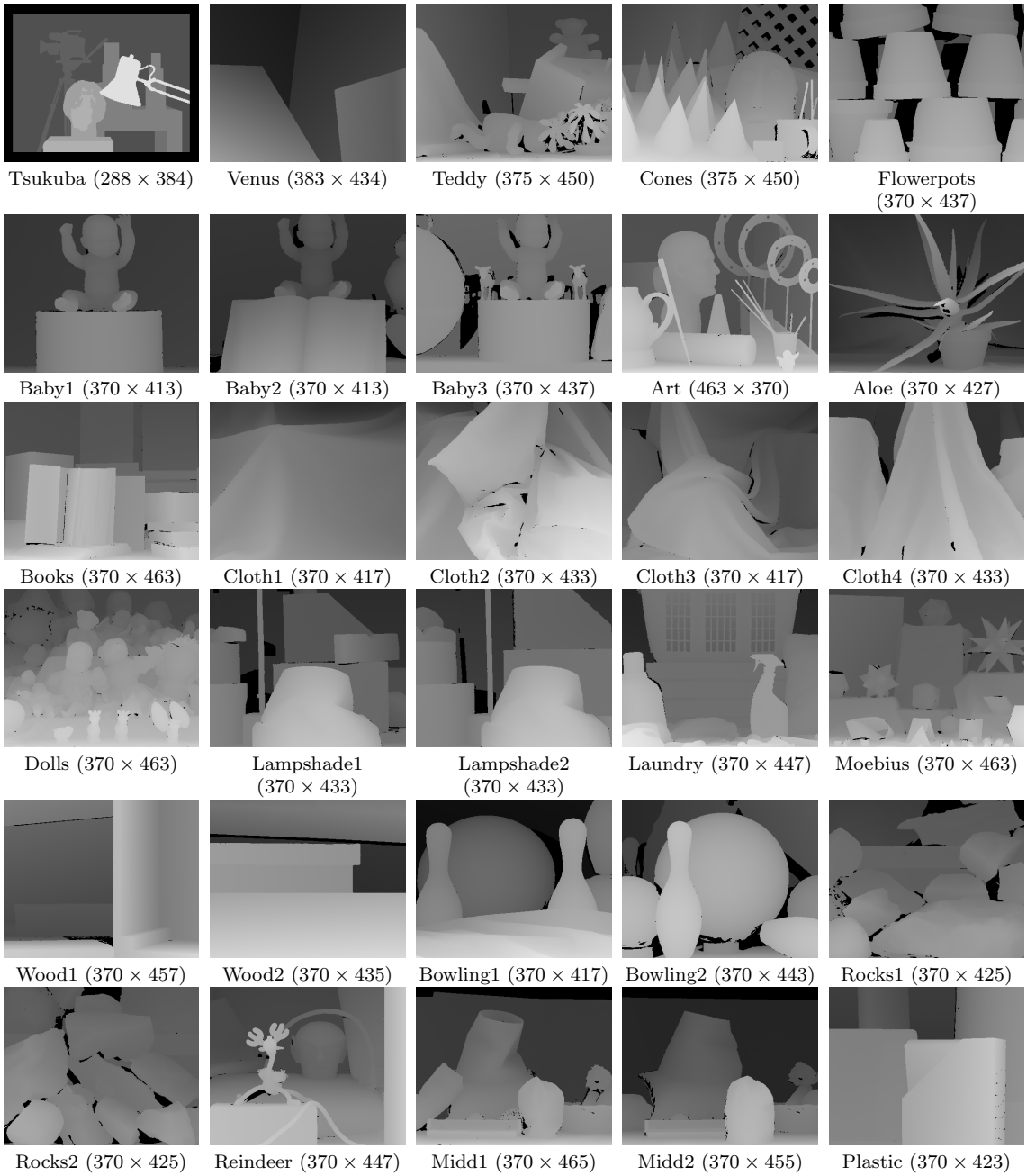


Figure 5.2: Ground truth disparity maps of 30 Middlebury data sets.

5.2 Parameter setting

The parameters for our proposed method are set empirically. Table 5.1 presents the parameter settings of our proposed non-local dense stereo matching algorithm in all the experiments. Note that the maximum disparity level d_{max} varies with the stereo image pairs, which is a part of given information of each data set.

Table 5.1: Parameter settings for proposed algorithm.

Step	Parameter	Value
Cost Computation	Weighting coefficient γ	0.3
	Weighting coefficient β	0.11
	Color truncation threshold T_i	7
	Gradient truncation threshold T_g	2
	HOG feature window size W	5
	Minimum disparity level d_{min}	1
	Maximum disparity level d_{max}	Depends on image
Non-local Cost Aggregation on Edge-aware T-MST	Edge weight truncation threshold τ	36
	Superpixel size S	300
	Smooth term σ	0.1
Adaptive Non-local Refinement	Truncation threshold τ_{cc}	$0.5 \cdot d_{max}$
	Weighting coefficient φ	0.1
	Smooth term σ	0.1

For the other five non-local stereo matching algorithms being compared in this chapter (MST, ST1, ST2, Cross-E, Cross-Sp), the parameters follow the settings of the corresponding papers [19, 20, 52, 60, 61].

5.3 Performance evaluation and comparison

In our experiments, we test the performance of our proposed method and other five non-local stereo matching algorithms on 30 Middlebury data sets. These 30 data sets consist of four standard Middlebury data sets which are commonly used for evaluation in almost all recent stereo matching algorithms, and other 26 newer Middlebury data sets. We will evaluate and compare the performance of the six non-local algorithms on standard Middlebury data sets first. Then more experimental results from the six methods will be presented.

5.3.1 Standard Middlebury data sets

Table 5.2 presents the numerical comparison of the our proposed method and other five non-local tree-based stereo matching algorithms on four standard Middlebury data sets (*Tsukuba, Venus, Teddy and Cones*). The numbers are the percentage of inaccurate pixels with error threshold 1 on different data sets. The bold number in each column indicates that the corresponding method has the most accurate result among all the methods for corresponding data set.

Table 5.2: Numerical comparison of the our proposed method and other five non-local tree-based stereo matching algorithms on four standard Middlebury data sets.

Method	Tsukuba			Teddy			Venus			Cones			Avg-
	non-occ	all	disc	non-occ	all	disc	non-occ	all	disc	non-occ	all	disc	error
MST	2.26	2.92	7.33	6.39	12.98	14.82	0.50	1.05	4.51	2.77	10.97	7.81	6.19
ST1	1.85	2.61	7.55	7.67	15.01	17.66	0.64	1.43	6.16	3.55	12.04	10.04	7.18
ST2	2.44	3.38	7.69	7.76	15.75	17.37	0.74	1.76	7.66	3.31	12.25	9.40	7.46
Cross-E	2.27	3.58	7.82	7.84	15.40	17.05	0.72	1.90	7.08	3.77	13.20	10.34	7.58
Cross-SP	2.30	3.55	9.31	7.88	15.34	18.18	0.61	1.74	7.26	3.18	12.28	9.21	7.57
Proposed	**1.59**	**2.18**	**7.25**	**5.42**	**10.47**	**12.74**	**0.18**	**0.48**	**1.87**	**2.61**	**9.25**	**7.78**	**5.15**

As we can see from Table 5.2, our proposed algorithm outperforms all other non-local tree-based stereo matching algorithms on all standard Middlebury data sets. The proposed algorithm shows outstanding robustness in three kinds of regions,which are non-occluded regions, all regions and discontinuous regions. Note that, the accuracy in non-occluded regions usually draws more attention than in other kinds of regions since only pixels in non-occluded regions actually have corresponding pixels exist in other image. The percentage of inaccurate pixels in discontinuous regions indicates how robust an algorithm is when dealing with the sharp disparity changes across depth boundaries. The percentage of inaccurate pixels in all regions shows the overall performance in terms of accuracy.

For *Tsukuba, Venus and Cones*, since the accuracy differences among six methods are not large enough to reliably distinguish the characteristics of each method, we only present the computed disparity maps of *Teddy*, as shown in Figure 5.3. Red pixels represent the inaccurate pixels in non-occluded regions with error threshold 1.

As we can see from Figure 5.3, computing accurate disparity values of pixels around the toy bear is a problem for all non-local tree based stereo matching algorithms, as shown in yellow rectangle. Although the proposed method still has some inaccurate pixels above and on the left side of the toy bear, the disparity values of pixels on the right side of the bear are accurately computed. MST and the proposed method have better performance in discontinuous regions as shown in blue rectangles, which can also be seen in Table 5.2. Green rectangle marks an occluded region where only exist in left image due to the shift of camera. The proposed algorithm performs well in this region, which shows that the proposed edge-aware T-MST is reliable for message passing.

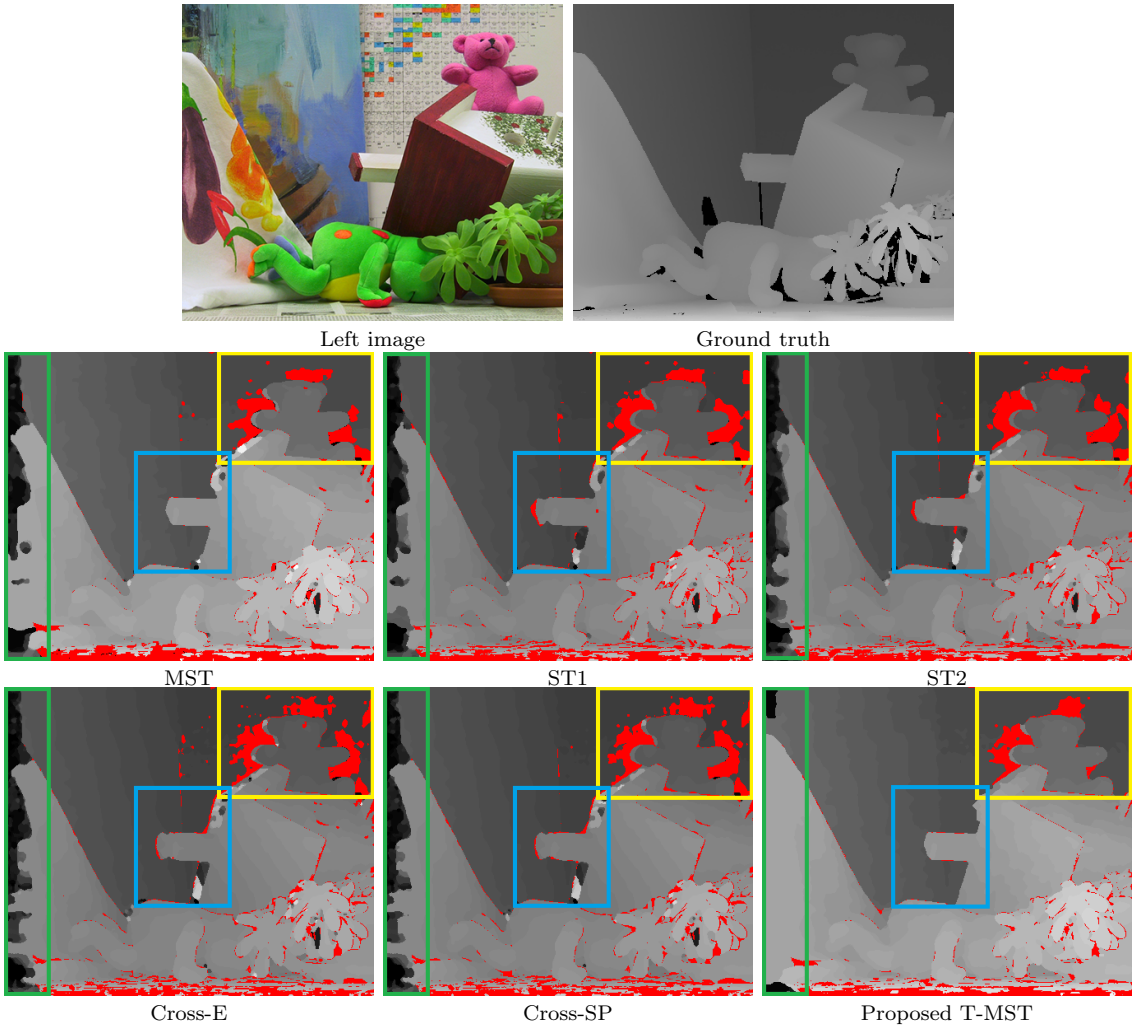


Figure 5.3: Final disparity maps on standard Middlebury data sets.

5.3.2 Other Middlebury data sets

Table 5.3 shows the quantitative accuracy evaluation of our proposed method and other five non-local methods on 30 image pairs in Middlebury. For each algorithm, the disparity maps are computed with the same framework and parameter settings as presented in the corresponding papers. Only the pixels in non-occluded regions

Table 5.3: Performance evaluation of the stereo matching accuracy.

Data	**MST**	**ST1**	**ST2**	**Cross-E**	**Cross-SP**	**T-MST**
Tsukuba	2.26_3	1.85_2	2.44_6	2.27_4	2.30_5	$\mathbf{1.59}_1$
Venus	0.50_2	0.64_4	0.74_6	0.72_5	0.61_3	$\mathbf{0.18}_1$
Teddy	6.39_2	7.67_3	7.76_4	7.84_5	7.88_6	$\mathbf{5.42}_1$
Cones	2.77_2	3.55_5	3.31_4	3.77_6	3.18_3	$\mathbf{2.61}_1$
Flowerpots	18.76_6	14.91_5	12.62_2	14.39_3	14.52_4	$\mathbf{12.21}_1$
Baby1	7.42_5	4.52_2	$\mathbf{4.07}_1$	4.68_3	4.69_4	9.54_6
Baby2	28.55_6	15.20_4	12.23_3	$\mathbf{6.33}_1$	6.38_2	15.74_5
Baby3	5.59_6	5.07_3	4.98_2	5.31_5	5.29_4	$\mathbf{4.31}_1$
Art	10.84_6	10.52_5	9.12_4	$\mathbf{8.58}_1$	8.80_3	8.64_2
Aloe	4.51_6	4.40_5	$\mathbf{3.67}_1$	4.02_3	3.68_2	4.29_4
Books	11.35_6	9.41_5	$\mathbf{8.17}_1$	8.26_2	8.35_3	8.37_4
Cloth1	0.35_5	0.41_6	0.31_4	0.26_3	$\mathbf{0.14}_1$	0.23_2
Cloth2	3.14_5	3.36_6	2.02_3	$\mathbf{1.80}_1$	1.87_2	2.20_4
Cloth3	1.53_4	1.57_5	1.39_3	1.67_6	1.26_2	$\mathbf{1.00}_1$
Cloth4	1.19_5	1.27_6	0.98_4	0.92_3	0.77_2	$\mathbf{0.72}_1$
Dolls	4.33_2	5.17_6	4.47_4	4.45_3	4.71_5	$\mathbf{3.56}_1$
Lampshade1	12.59_6	10.31_3	9.90_2	10.47_4	10.67_5	$\mathbf{8.79}_1$
Lampshade2	16.60_4	21.65_6	19.36_5	14.70_2	15.03_3	$\mathbf{8.44}_1$
Laundry	11.32_2	13.83_4	13.30_3	14.30_5	14.37_6	$\mathbf{10.57}_1$
Moebius	8.83_6	8.19_5	$\mathbf{7.53}_1$	8.16_3	8.01_2	8.17_4
Wood1	13.57_6	5.08_4	$\mathbf{3.58}_1$	3.72_2	3.82_3	12.72_5
Wood2	3.21_6	3.06_5	2.13_4	$\mathbf{0.88}_1$	0.98_2	1.71_3
Bowling1	27.16_6	18.88_4	16.43_3	15.53_2	$\mathbf{15.52}_1$	21.38_5
Bowling2	13.53_6	11.13_5	$\mathbf{7.83}_1$	8.80_2	8.85_3	11.02_4
Rocks1	1.88_5	2.34_6	1.85_4	1.60_2	1.76_3	$\mathbf{1.36}_1$
Rocks2	1.35_3	1.61_6	1.37_4	1.27_2	1.44_5	$\mathbf{1.16}_1$
Reindeer	8.34_5	7.73_4	5.68_2	5.73_3	$\mathbf{5.60}_1$	10.23_6
Midd1	20.74_2	31.37_3	33.93_4	35.74_6	35.55_5	$\mathbf{7.09}_1$
Midd2	48.13_6	28.67_2	34.27_5	31.76_3	31.87_4	$\mathbf{6.87}_1$
Plastic	54.36_6	40.82_5	33.95_2	36.81_4	36.59_3	$\mathbf{30.04}_1$
Avg. error	11.72_6	9.81_5	8.98_4	8.82_2	8.82_2	$\mathbf{7.34}_1$
Avg. rank	4.67_6	4.47_5	3.10_4	3.17_2	3.23_3	$\mathbf{2.37}_1$

are taken into consideration during the calculation of quantitative accuracy evaluation. For more comprehensive comparison, the evaluation results of four standard Middlebury data sets are also included in Table 5.3 (row 2 to row 5).

In Table 5.3, the normal numbers in row 2 to row 31 are the percentages of inaccurate pixels in non-occluded regions with error threshold 1 for each data set. The subscript numbers are the relative rank for each data set. The last two rows show the average error rates and average ranks of the six methods. Table 5.3 reveals some important characteristics of the six non-local stereo matching algorithms about their performances.

First, MST outperforms ST1, ST2, Cross-E and Cross-SP in standard Middlebury data sets in almost all cases. However, when more data sets are included in evaluation, MST is less accurate. ST1 outperforms MST in 17 data sets; ST2 outperforms MST in 21 data sets; Cross-E outperforms MST in 22 data sets; Cross-SP outperforms MST in 22 data sets. It can be proved that performing non-local cost aggregation over segment-trees and cross-trees with prior does improve the accuracy of computed disparity maps.

Second, conducting non-local cost aggregation over cross-trees with prior generally produces more accurate disparity maps than over segment-trees. Cross-E and Cross-SP outperform ST1 in 19 and 22 data sets respectively; outperform ST2 in 14 and 15 data sets respectively. The differences between the performances of Cross-E and Cross-SP are very narrow in most data sets, which shows that edge-prior and superpixel-prior can both provide reliable depth boundary detection.

Third, the proposed algorithm consistently outperforms all other five non-local tree-based stereo matching algorithms. T-MST outperforms MST in 28 data sets;

outperforms ST1 in 25 data sets; outperforms ST2 in 20 data sets; outperforms Cross-E in 18 data sets; outperforms Cross-Sp in 18 data sets. Among all the 30 data sets, the proposed method achieves the most accurate results for 17 data sets.

Finally, comprehensive comparison and ranking of the six methods can be drawn from the last two rows of Table 5.3: T-MST > Cross-SP ≈ Cross-E > ST2 > ST1 > MST, where ">" means "outperform" and "≈" means "almost equivalent". The proposed method has the least average error and the highest average ranking among the six methods, which demonstrates that the proposed method has better performance in terms of accuracy in a comprehensive way.

Visual comparisons of our proposed method and five non-local tree-based stereo matching methods for three Middlebury data sets, *Laundry* (Figure 5.4), *Lampshade1* (Figure 5.5) and *Midd2* (Figure 5.6), are presented. Regions marked in red show the inaccurate pixels with error threshold 1 in non-occluded regions of final disparity maps.

Figure 5.4 shows the final disparity maps computed with six non-local algorithms on data set *Laundry*. The main scene objects in this data set are: big white laundry basket (upper-middle); wooden box under the laundry basket (bottom-middle); red detergent bottle (bottom-left); white spray bottle and green stripe pillow (bottom-right); clothes (bottom).

As we can see from Figure 5.4, the proposed method performs well in the wooden box (marked with yellow rectangle), even in the gaps between woods, while other methods compute inaccurate disparity values in those gaps. In addition, the disparity boundaries around the spray bottle are preserved well in final result of the proposed method (marked with blue rectangle), while ST1, ST2, Cross-E and Cross-SP fail

to preserve clear disparity boundaries. The performance in this data set shows that our proposed algorithm is able to provide reliable disparity results around disparity boundaries. There are two main reasons for the errors in the holes of the laundry basket: prior is detected and the intensity differences are large, which all suppress the aggregation flow in these regions.

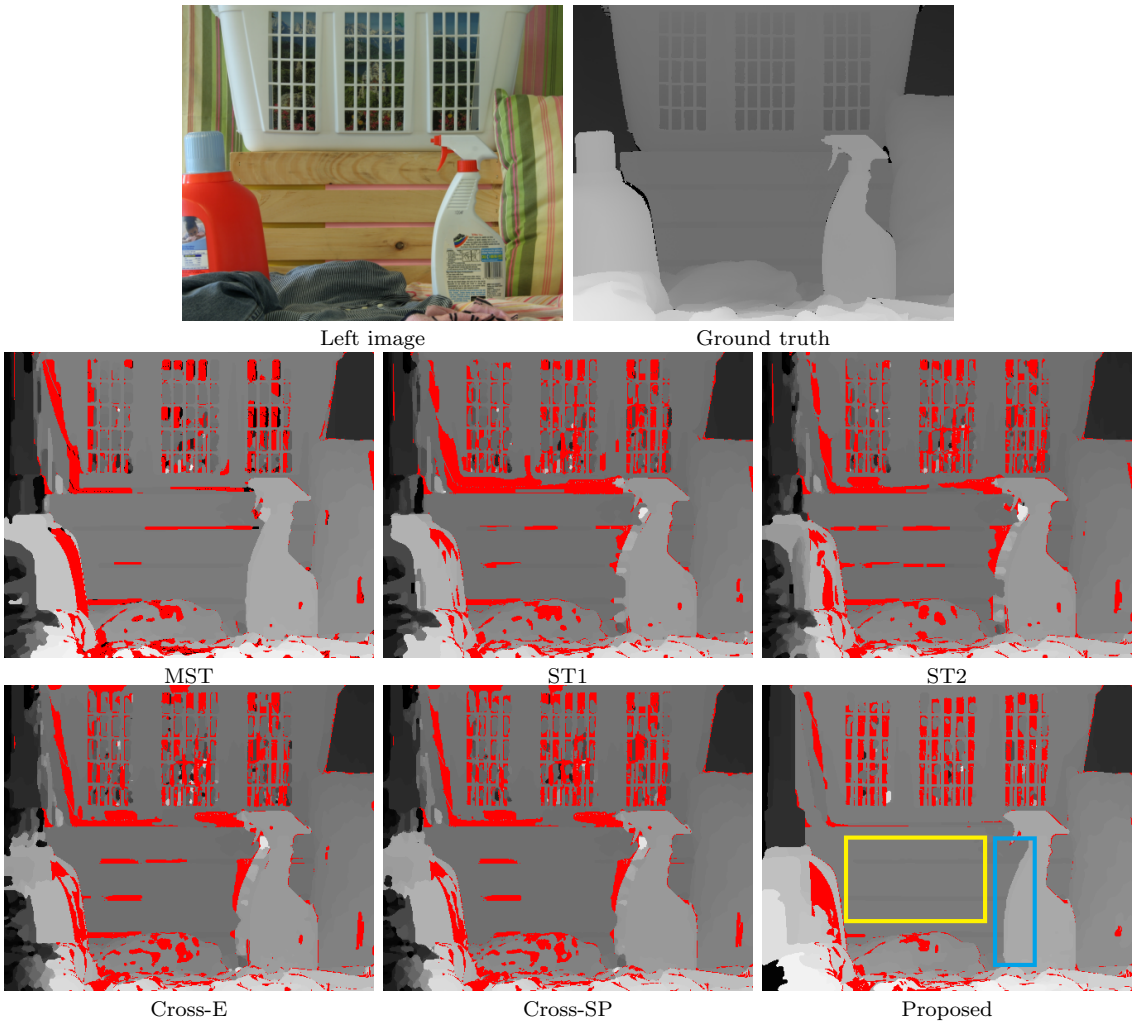


Figure 5.4: The final disparity maps of Laundry.

Figure 5.5 shows the final disparity maps computed with six non-local algorithms

on data set *Lampshade1*. The main scene objects in this data set are: white lampshade (bottom-middle); carton behind the lampshade (middle); wood brick and yellow magazine file on the carton (upper-middle); wooden pole (left-middle); two round boxes and a wood brick (left).

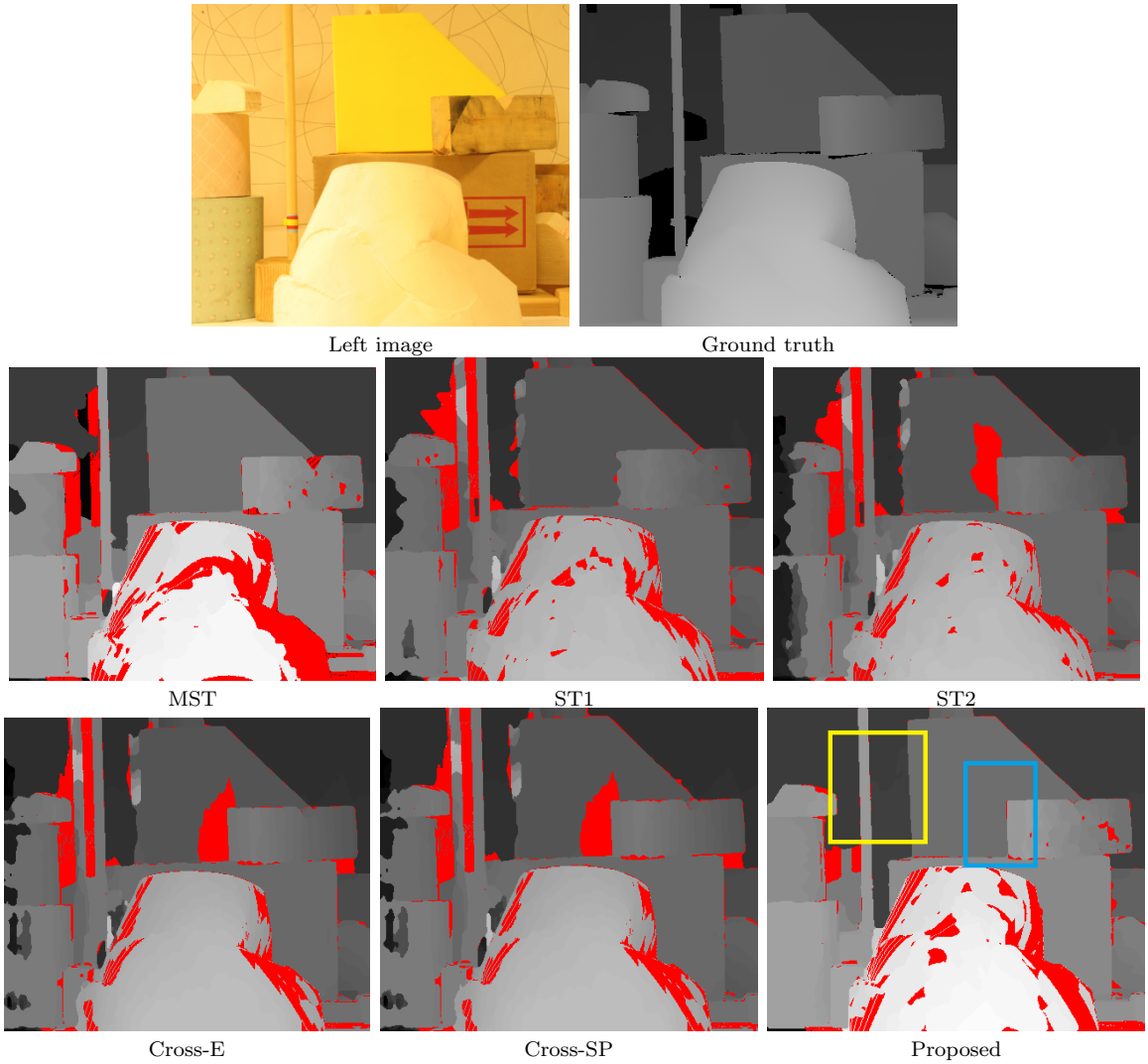


Figure 5.5: The final disparity maps of Lampshade1.

As we can see from Figure 5.5, the proposed algorithm performs well in preserv-

ing the disparity boundaries (marked with yellow and blue rectangles). Our proposed method performs better than other five methods within large textureless planar surfaces because the cost aggregation in those regions are enforced due to the truncated edge weight when the intensity differences are small and no hybrid edge prior is detected.

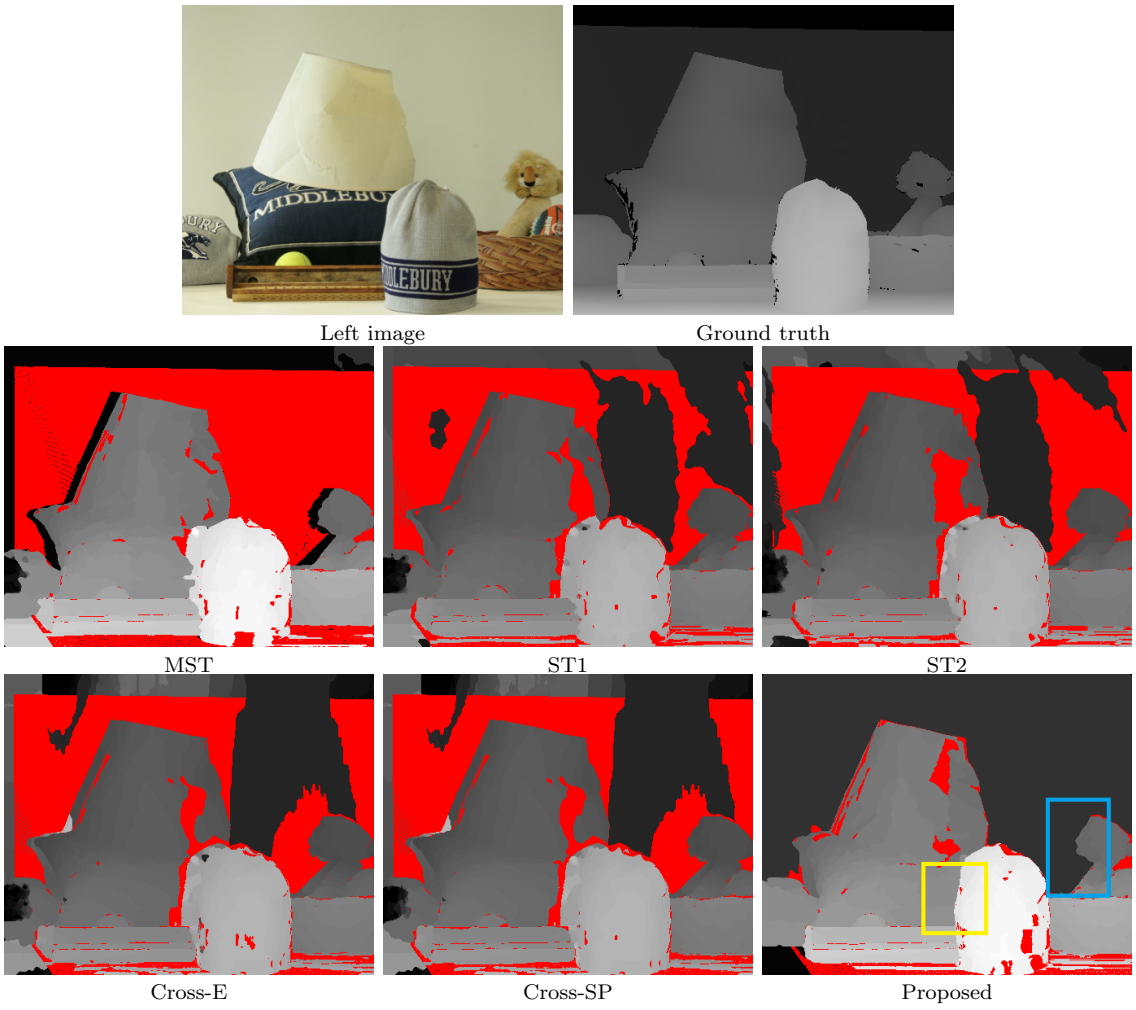


Figure 5.6: The final disparity maps of Midd2.

Figure 5.6 presents the final disparity maps of the six non-local algorithms on

the data set *Midd2*. The main scene objects in this data set are: folded T-shirt (bottom-left); blue pillow (middle); white lampshade on the pillow (middle); wood and tennis ball (bottom); grey woollen hat (bottom); woven basket and toys (right). *Midd2* is a very challenging data set for local stereo matching algorithms due to large repeated textures. In such case, several potential corresponding pixels in the other image can be detected for each pixel in the reference image because local methods find correspondences locally based on pixel similarities. The inaccurate pixel rate in non-occluded regions for MST, ST1, ST2, Cross-E and Cross-SP are all above 28 on this data set (row 30 in Table 5.3). The proposed algorithm reduces the error rate to 6.87, which is an incredible improvement.

As we can see from Figure 5.6, the proposed method performs the best among all six non-local tree-based stereo matching algorithms in large textureless planar background. Almost all disparities in background are accurate in disparity map computed with our proposed method. In addition, disparity boundaries are preserved with the proposed method since the false cost aggregation across depth boundaries are suppressed by hybrid edge-prior (marked with yellow and blue rectangles).

5.4 Summary

In this chapter, we tested our proposed non-local stereo matching method based on edge-aware T-MST and compared with five state-of-the-art non-local methods on 30 Middlebury data sets. The experiments are separated as two parts: in the first part, we analysed and compared the performances of six non-local algorithms on four standard Middlebury data sets in terms of accuracy. Quantitative evaluations in non-

occluded regions, all regions and discontinuous regions were presented in this part, which show that our proposed algorithm outperforms other five methods in all three kinds of evaluation regions on standard Middlebury data sets.

In the second part, in order to obtain a more comprehensive understanding, we further evaluated and compared our proposed method with other methods on 26 more Middlebury data sets. The experiments showed that our proposed algorithm consistently outperforms other methods, especially in large textureless planar regions and disparity boundaries due to the proposed edge-aware T-MST.

In conclusion, comparisons on quantitative evaluations and visual qualities of the performances of our proposed method and other five non-local methods on 30 Middlebury data sets are presented in this chapter. The comparison demonstrates that our proposed non-local stereo matching algorithm based on edge-aware T-MST outperforms the current state-of-the-art non-local tree-based stereo matching methods.

Chapter 6

Conclusions

Extracting accurate depth maps from stereo image pairs is an important requirement for many applications such as 3D reconstruction. A large variety of algorithms have been proposed to perform stereo matching. Some techniques generate a sparse depth map by matching reliable features, while the more common algorithms seek dense corresponding pairs between stereo images.

Typically, dense stereo matching algorithms can be further divided into two broad categories: local methods and global methods. In this thesis, a novel dense non-local tree-based stereo matching algorithm, which balances the accuracy of global methods and the speed of local method, has been proposed.

One of the important challenges for aggregation-based stereo matching algorithm is how to find optimal support regions. Traditional local methods only aggregate within a fixed window, while the non-local algorithm generates a tree structure which contains all pixels in the reference image so that the support region for each pixel is the whole image. Performing aggregation on such tree structure allows each pixel to

receive weighted supports from all other pixels. In our scheme, a novel edge-aware T-MST is proposed for conducting non-local cost aggregation. In our proposed edge-aware T-MST, truncated edge weights are employed to enforce strong cost aggregations. Meanwhile, a hybrid edge prior is proposed to suppress false cost aggregations across disparity boundaries and preserve true depth boundaries.

In addition, a novel matching cost computation function, which is robust to outlier pixels and illumination variation, is proposed in this thesis. We also present an adaptive non-local refinement scheme based on pixel stabilities, so that a more reliable final disparity map can be achieved.

The experimental results on Middlebury data sets show that the proposed algorithm successfully produces reliable disparity values within large planar textureless regions and around object disparity boundaries. Performance comparisons between the proposed algorithm and the other five non-local algorithms demonstrate that the proposed non-local stereo matching algorithm outperforms current state-of-the-art non-local tree-based stereo matching methods in most cases, especially in large textureless planar regions and around disparity bounaries.

However, limitations of our proposed scheme still exist: since the non-local cost aggregation depends on intensity similarities, the proposed scheme may fail in large homogeneous intensity regions with non-linear disparity changes. More work will be done in our future research. We are trying to compute the maximum a posteriori (MAP) disparity for each pixel using message-passing scheme on hidden Markov tree, which deals with the existing drawback of our proposed method.

References

[1] Robert F van der Willigen, Wolf M Harmening, Sabine Vossen, and Hermann Wagner. Disparity sensitivity in man and owl: Psychophysical evidence for equivalent perception of shape-from-stereo. *Journal of vision*, 10(1):1–11, 2009.

[2] Charles Wheatstone. Contributions to the physiology of vision. –part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical transactions of the Royal Society of London*, 128:371–394, 1838.

[3] Eisaku Katayama, Tomoko Shiraishi, Kenji Oosawa, Norio Baba, and Shin-Ichi Aizawa. Geometry of the flagellar motor in the cytoplasmic membrane ofsalmonella typhimuriumas determined by stereo-photogrammetry of quick-freeze deep-etch replica images. *Journal of Molecular Biology*, 255(3):458–475, 1996.

[4] O Taconet and V Ciarletti. Estimating soil roughness indices on a ridge-and-furrow surface using stereo photogrammetry. *Soil and Tillage Research*, 93(1):64–76, 2007.

[5] T Sarjakoski. Concept of a completely digital stereo plotter. *The Photogrammetric Journal of Finland*, 8(2):95–100, 1981.

[6] C Hernandez Esteban and Francis Schmitt. Multi-stereo 3d object reconstruction. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 159–166, 2002.

[7] Liang Zhang. Fast stereo matching algorithm for intermediate view reconstruction of stereoscopic television images. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(10):1259–1270, 2006.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[9] Suresh B Marapane and Mohan M Trivedi. Region-based stereo analysis for robotic applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1447–1464, 1989.

[10] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[11] Sébastien Roy and Ingemar J Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *IEEE International Conference on Computer Vision*, pages 492–499, 1998.

[12] Jongchul Lee, Daeyoon Jun, Changkyoung Eem, and Hyunki Hong. Improved census transform for noise robust stereo matching. *Optical Engineering*, 55(6):1–10, 2016.

[13] James K. Archibald Wade S. Fife. Improved census transforms for resource-optimized stereo vision. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):60–73, 2012.

[14] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158, 1994.

[15] Daniel Scharstein Heiko Hirschmuller. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599, 2008.

[16] Jingyi Yu Jingdan Zhang, L. McMillan. Robust tracking and stereo matching under variable illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–22, 2006.

[17] Xueguang Dong Jiling Liu, Yong Zhang. Local stereo matching based on the improved matching cost function and the adaptive window. In *International Congress on Image and Signal Processing*, pages 14–16, 2015.

[18] Bongjoe Kim Seungryong Kim, Bumsub Ham. Mahalanobis distance cross-correlation for illumination-invariant stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1844–1859, 2014.

[19] Qingxiong Yang. A non-local cost aggregation method for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1402–1409, 2012.

[20] Yang, Qingxiong. Stereo matching using tree filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):834–846, 2015.

[21] Eric T Psota, Jedrzej Kowalczuk, Mateusz Mittek, and Lance C Perez. Map disparity estimation using hidden markov trees. In *IEEE International Conference on Computer Vision*, pages 2219–2227, 2015.

[22] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[23] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515, 2001.

[24] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *IEEE International Conference on Computer Vision*, pages 900–906, 2003.

[25] Olga Veksler. Stereo correspondence by dynamic programming on a tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–390, 2005.

[26] Yuichi Ohta and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[27] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using

belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[28] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1458–1465, 2010.

[29] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewénius, and David Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.

[30] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[31] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. Map estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[32] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[33] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814, 2005.

[34] Bing Ma, Alfred Hero, John Gorman, and Olivier Michel. Image registration with minimum spanning tree algorithm. In *International Conference on Image Processing*, volume 1, pages 481–484, 2000.

[35] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[36] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[37] Richard Szeliski. Prediction error as a quality metric for motion and stereo. In *IEEE International Conference on Computer Vision*, volume 2, pages 781–788, 1999.

[38] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[39] Yingyun Yang Bo Liu, Qian Liang. An adaptive window stereo matching based on seed voting. In *IEEE International Conference on Signal Processing, Communications and Computing*, pages 771–774, 2014.

[40] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 103–110, 2001.

[41] Sergiu Nedevschi Mircea Paul Muresan, Mihai Negru. Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint. In

IEEE International Conference on Intelligent Computer Communication and Processing, pages 179–185, 2015.

[42] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006.

[43] Changyu Diao Jingzhou Huang. Adaptive support weight aggregation in segmentations for stereo matching. In *IEEE International Conference on Computer and Computational Sciences*, pages 292–296, 2015.

[44] Daniel Scharstein Richard Szeliski, Ramin Zabih. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

[45] Cyril Cassisa. Local vs global energy minimization methods: Application to stereo matching. In *IEEE International Conference on Progress in Informatics and Computing*, pages 678–683, 2011.

[46] V. Kolmogorov Y. Boykov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[47] Aaron F Bobick and Stephen S Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[48] Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and Neil A Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-

bilateral grid. In *European Conference on Computer Vision*, pages 510–523, 2010.

[49] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *European Conference on Computer Vision*, pages 1–14, 2010.

[50] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024, 2011.

[51] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2013.

[52] Xing Mei, Xun Sun, Weiming Dong, Haitao Wang, and Xiaopeng Zhang. Segment-tree based cost aggregation for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 313–320, 2013.

[53] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[54] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[55] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *IEEE International Conference on Computer Vision*, pages 1154–1160, 1998.

[56] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[57] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[58] Vladimir Kolmogorov, Antonio Criminisi, Andrew Blake, Geoffrey Cross, and Carsten Rother. Bi-layer segmentation of binocular stereo video. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 407–414, 2005.

[59] Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. Fusion of geometry and color information for scene segmentation. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):505–521, 2012.

[60] Feiyang Cheng, Hong Zhang, Mingui Sun, Helong Wang, and Ding Yuan. Cross-trees for stereo matching with priors. In *International Conference on Pattern Recognition*, pages 208–213, 2014.

[61] Feiyang Cheng, Hong Zhang, Mingui Sun, and Ding Yuan. Cross-trees, edge and superpixel priors-based cost aggregation for stereo matching. *Pattern Recognition*, 48(7):2269–2278, 2015.

[62] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[63] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 195–202, 2003.

[64] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[65] Xu Huang, Yongjun Zhang, and Zhaoxi Yue. Image-guided non-local dense matching with three-steps optimization. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3):67–74, 2016.

[66] Dung T Vu, Benjamin Chidester, Hongsheng Yang, Minh N Do, and Jiangbo Lu. Efficient hybrid tree-based stereo matching with applications to postcapture image refocusing. *IEEE Transactions on Image Processing*, 23(8):3428–3442, 2014.

[67] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[68] Cheng Lei and Yee-Hong Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *IEEE International Conference on Computer Vision*, pages 1562–1569, 2009.

[69] Jiangbo Lu, Hongsheng Yang, Dongbo Min, and Minh N Do. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1854–1861, 2013.

[70] Daniel Scharstein and Richard Szeliski. Middlebury stereo matching datasets. `http://vision.middlebury.edu/stereo/data/`. Accessed: 2016-09-15.

[71] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[72] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.