OCCLUSION AWARE STEREO MATCHING WITH O(1) COMPLEXITY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YETI ZIYA GÜRBÜZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2014

Approval of the thesis:

OCCLUSION AWARE STEREO MATCHING WITH O(1) COMPLEXITY

submitted by **YETI ZIYA GÜRBÜZ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Eng.**

Prof. Dr. A. Aydın Alatan
Supervisor, **Elec. and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Department, METU

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Department, METU

Asst. Prof. Dr. Fatih Kamışlı
Electrical and Electronics Engineering Department, METU

Prof. Dr. Levent Onural
Electrical and Electronics Eng. Dept., Bilkent University

Dr. Cevahir Çığla
SST, ASELSAN

Date: September 05, 2014

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: YETI ZIYA GÜRBÜZ

Signature :

ABSTRACT

OCCLUSION AWARE STEREO MATCHING WITH O(1) COMPLEXITY

Gürbüz, Yeti Ziya
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. A. Aydın Alatan

September 2014, 146 pages

The problem of joint reduction of computational complexities of local stereo matching methods due to both cost aggregation step and correspondence search range is addressed and a novel hierarchical stereo matching algorithm is presented. The proposed approach exploits edge aware recursive volume filtering with a reduction on correspondence search range. The fastest state-of-the-art edge aware recursive filters are modified so that they become applicable to the methods to reduce the complexity in correspondence search range. In this way, complexities due to both cost aggregation step and correspondence search range are eliminated, yielding an O(1) complexity algorithm. In addition, the weakness of recursive filters in the presence of noise or high texture is handled by the help of a hierarchical scheme. Unlike common hierarchical methods, the transfer of the disparity estimates across the scales is converted into an optimization problem in order to preserve object boundaries and propagate proper estimates across scales. Dynamic programming is exploited to solve this optimization problem efficiently. The proposed transfer method can be utilized to transfer disparity across scales either in an image pyramid between stereo pairs or along frames in stereo video. The occlusion problem is solved inherently by the proposed approach that provides further decrease in complexity. The experimental results show that the proposed method provides quite efficient computation for stereo matching with a marginal decrease in performance. Compared to the state-of-the-art techniques, the proposed technique is possibly the fastest approach with a comparable accuracy based on benchmarking with Middlebury stereo pairs.

Keywords: Stereo matching, cost volume filtering, hierarchical disparity transfer, asymmetric occlusion handling, O(1) complexity

ÖZ

ÖRTMELERİ GÖZETEREK O(1) KARMAŞIKLIKTA STEREO EŞLEME

Gürbüz, Yeti Ziya
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. A. Aydın Alatan

Eylül 2014 , 146 sayfa

Yerel stereo eşleme yöntemlerinin hem maaliyet kümelenmesi adımından hem de ilişkilendirme arama aralığından kaynaklanan işlem karmaşıklığını aynı anda azaltma sorunu ele alınmıştır ve yeni bir sıradüzensel stereo eşleme yöntemi sunulmuştur. Önerilen yöntem, özyinelemeli ayrıt koruyan süzgeçlemeden faydalanarak ilişkilendirme arama aralığını azaltıp hacimsel süzgeçleme yapmaktadır. Literatürde önde gelen en hızlı özyinelemeli ayrıt koruyan süzgeçler, ilişkilendirme arama aralığından kaynaklanan karmaşıklığı azaltmaya yönelik yöntemlere uygun hale getirilecek şekilde geliştirilmiştir. Bu şekilde hem maaliyet kümelenmesi adımından hem de ilişkilendirme arama adımından kaynaklanan işlem karmaşıklıkları elenerek O(1) karmaşıklıkta bir yöntem oluşturulmuştur. Aynı zamanda, sıradüzensel yapı sayesinde özyinelemeli süzgeçlerin gürültü ve yüksek doku karşısındaki zayıflıkları giderilmiştir. Genel sıradüzensel yöntemlerin aksine, farklı çözünürlükler arasındaki ayrıklık kestirimi aktarımı, nesne ayrıtlarını korumak ve uygun ayrıklık kestirimilerinin aktarımını sağlamak amacıyla bir en iyileme problemine çevrilmiştir. Bu en iyileme problemini verimli şekilde çözebilmek için dinamik programlama kullanılmıştır. Önerilen ayrıklık kestirimi aktarma yöntemi, imge piramidindeki farklı çözünürlüklü görüntüler arasındaki ayrıklık kesitirimlerinin aktarımı için kullanılabileceği gibi stereo eşleri ya da bir stereo videodaki film kareleri arasındaki ayrıklık aktarımı için de kullanılabilir. Şunu belirtmek gerekir ki örtme sorunu önerilen yaklaşım ile içsel olarak çözülmüştür. Bu sayede işlem karmaşıklğında daha da azalma sağlanmıştır. Deneysel sonuçlar önerilen yöntemin stereo eşleme için başarımda az bir düşüş ile son derece verimli bir heseaplama sağladığını göstermiştir. Önerilen yöntem, Middlebury stereo görüntü verileri temel alındığında, önde gelen yöntemler ile benzer başarımda olan muhtemel en hızlı yöntemdir.

Anahtar Kelimeler: Stereo eşleme, hacim süzgeçleme, hiyerarşik ayrıklık aktarımı, asimetrik örtme çözme, O(1) karmaşıklık

To:

My family, my grandma, love of my life and my brother...

ACKNOWLEDGMENTS

I inform that there have been many people around me during my graduate studies and these are the people who all make it happen for me to present a thesis study I am comfortable with. Unfortunately, only few of them can be mentioned here due to lack of space.

First and foremost, I would like to express my sincere thanks to my supervisor, Prof. Dr. A. Aydın Alatan for his constant support and trust in me. I would not be aware of many research areas without his guidance and leading. In addition to performing a research work, how to express an academic topic so simply that even my grandma can understand and how to conduct a research and development project are only few of the things I have got from him. I am grateful for his effort to provide me with extra funds via research and development projects. He has not only made my life more comfortable and suitable for research but also given me opportunity to be productive in my very early academic life. I have always felt privileged to be a graduate student of such a brilliant person and be one of the members of his research laboratory.

Secondly, as it has been just mentioned, I also express my gratitude to current and past member of Multi Media Research Laboratory. I thank Dr. Cevahir Çığla for leaving me a legacy of a great study he has performed during his doctoral studies and giving me the opportunity to improve it by encouraging me to dig it more. I offer special thanks to Emrecan Batı for his teachings and assistance about almost everything, listening me patiently every time I would like to talk about my progress in my studies and the academic discussions we have made. I thank Emin Zerman for encouraging me by mentioning his trust in my knowledge on every occasion and not only serving as a model of a disciplinary person but also enforcing me to be so. I thank Beril Beşbınar for her partnership for overcoming the obstacles present in the research project we get involved in

together and also making all of us at home in the laboratory. I also thank Erhan Gündoğdu for sharing his know-how and experience with me for my studies. I give my special thanks to İlker Buzcu for making it possible to finish my thesis study two months earlier(!) and Ömurcan Kumtepe for being a kind companion during the teaching assistant works we have conducted together.

Most importantly, I offer my greatest and sincere thanks to love of my life, Melis Özateş, for her support in both personal and academic life. She makes me rise to my feet every time I fall by believing me so much like nobody ever does. She has been with me all the time with patience and love. Especially, when I was stuck, her support and encouragement triggered my enthusiasm to move on towards finishing my studies. Additionally, expanding my vision towards optimization world and her discussions about statistics are just a few of the many contributions of her to my academic life. There is no doubt that without her I would not be where I am today.

Moreover, no thank would be enough for my family's support. My parents, Nesrin - Başar Gürbüz, and my sister, Börte Gürbüz Özgür, have made everything all possible with their endless support. They have put their ultimate effort into my academic life since the very first day and always reminded me of not worrying about anything else but my studies. I also inform that the greatest of the thanks goes to my grandma for encouraging me to move on in accordance with my purposes. Finally, there is actually no word expressing how thankful I am for my brother, my dog, Dobi Gürbüz, for being my best friend all the time. I had always felt the most privileged to have grown up with him. He showed me very different aspect of love.

Last, but not least, I would love to give my sincere thanks to TÜBİTAK BİDEB for the MSc scholarship enabling me to focus on academic studies more rather then distracting myself about funding.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BP	Belief propagation
CT	Census transform
CTF	Coarse-to-fine
DP	Dynamic programming
DSI	Disparity space image
EAF	Edge-aware filtering
GC	Graph cut
LRC	Left-right checking
MC	Matching cost
METH	Minimum error thresholding
MRF	Markov random field
REAF	Recursive edge-aware filtering
RIP	Recursive information prediction
SAD	Sum of absolute differences
WTA	Winner take all

CHAPTER 1

INTRODUCTION

Stereo image pairs are the result of capturing the same scene from two distinct vantage points with a camera. Relative positions of the projections of the objects in the scene onto the image planes enable extraction of depth information. This process can be considered as *stereo vision.* Depth estimation from stereo image pairs has been in researchers field of interest for more than three decades [52] and has always maintained its popularity. It is still a hot topic in computer vision due to not only increasing demand for depth perception but also stereo camera's advantages over other depth sensors.

Still being a hot topic in computer vision comes from not only increasing demand for depth perception but also stereo camera's advantages over other depth sensors.

Advances in autonomous systems and entertainment business have increased the attention on depth extraction of 3-D scenes; since, gathering or having the depth information of a scene is the integral part of such computer vision systems as autonomous driving, human motion analysis, augmented reality, content generation for 3-D and free-view TVs. Depth information can be acquired in several ways using different sensors. Among those, stereo vision with cameras is more attractive because of two main reasons. Firstly, it is easier to be accessed; having only two cameras is enough for a stereo vision sensor. Secondly, camera is a passive sensor; namely, for its readings, it does not emit any form of energy to the environment but uses the radiation available instead. This makes it more preferable when the environment is not desired to be influenced and makes it more robust to different environments like indoors and outdoors. Moreover, high resolution depth maps can be obtained from stereo cameras and in addition to depth information, camera images convey very rich information of the environment as well. Benefits of stereo vision together with need for depth perception pique researchers' interest in depth estimation from stereo image pairs.

Depth estimation techniques from stereo image pairs can be mainly classified into two as sparse and dense. Sparse methods work on distinctive image features such as corners, blobs, etc. and estimate depth only at that pixels, resulting sparse depth maps; whereas, dense methods work on every pixel of the image and estimate depth for every pixel. Many studies have been performed and sophisticated algorithms have been developed especially for dense depth estimation [63]; since, many modern applications of stereo vision require dense depth maps. Currently, state-of-the-art methods can produce high quality depth maps from stereo image pairs. However, when it comes to using stereo vision in real-time systems, many of the superior methods cannot be applicable due to their computational complexities. Therefore, to develop computationally efficient methods with close performance to state-of-the-art is the current trend in stereo vision.

1.1 Problem Definition

Once relative positions of two cameras and corresponding locations of the projections of a 3-D point onto image planes[1] are known, depth information of that point can be extracted from stereo image pairs provided that cameras can be approximated by the *pinhole camera model*[2]. This situation is depicted in Fig. 1.1. *Center of projections* of two cameras are located at $\bar{c}_l$ and $\bar{c}_r$ with a relative rotation and translation, (R, t). A point, $\bar{P}$, in the scene is projected onto virtual image planes at locations $\bar{p}_l$ and $\bar{p}_r$ for the left and the right camera, respectively. Left image imposes a constraint on possible 3-D locations of point $\bar{P}$ according to perspective projection. This constraint is the half line, d_l, extending from $\bar{c}_l$

[1] The term *image plane* does not mean actual image plane inside the camera. It is used to refer *virtual image plane* in this thesis instead.

[2] [69] can be referred for image formation in pinhole cameras.

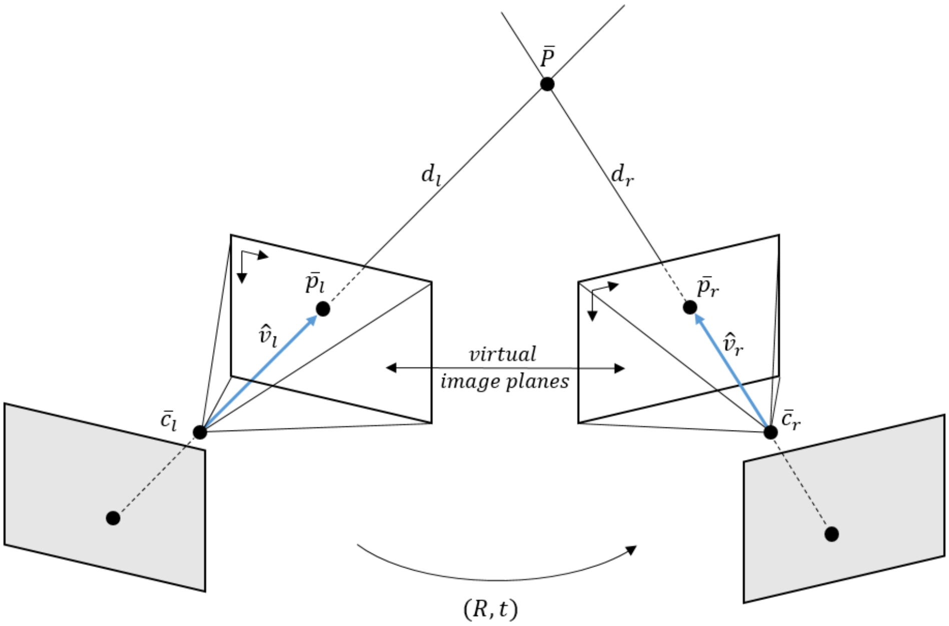


Figure 1.1: Triangulation. Depth information of a scene point relative to image planes can be extracted from its projections on the image planes of two cameras as long as the relative positions of the cameras are known.

and passing through $\bar{p}_l$. A similar constraint comes from right image as well. These two constraints together reduce the possible 3-D locations of point $\bar{P}$ to a single point, enabling determination of depth of the point relative to cameras. It can be concluded from Fig. 1.1 that when relative camera positions are given, the key to extract depth of a point is to know the projections of that point in both images. Therefore, depth extraction problem can be considered as search for the correspondence of a location in one image in the other image. Similarly, for digital images, it is matching a pixel[3] in one image with a pixel in the other image so that pixels in matched pairs are the projections of the same point.

Given a pixel in one image, its correspondence/match in the other image should be determined in order to extract the depth information. This requires a search in the domain on which image is defined. *Epipolar geometry*, which can be considered as the geometry of stereo vision, allows restriction on the space where the correspondence search is to be performed by introducing geometric constraints. There are three important components of *epipolar geometry*. First one

[3] Throughout this thesis, the terms *pixel* and *image point* can be used interchangeably.

is *epipolar plane* which is the plane passing through center of projection of the left and right cameras, $\bar{c}_l$ and $\bar{c}_r$, and a point in the scene, $\bar{P}$. Second is *epipolar line* which is the line formed by the intersection of *epipolar plane* and image plane. Last one is *epipolar point* or *epipole* which is the point where center of projection of one camera projects onto the other camera's image plane. These points are denoted by $\bar{e}_l$ and $\bar{e}_r$ and illustrated in Fig. 1.2 together with the other components. By examining Fig. 1.2, some important observations can be made. Every *epipolar plane* passes through *epipoles* regardless of the location of point $\bar{P}$. An *epipolar line* in one image plane passes through the *epipole* and the projection of the scene point defining the *epipolar plane* together with center of projection of the cameras onto that image plane. Namely, point $\bar{P}$, its projections onto image planes, $\bar{p}_l$ and $\bar{p}_r$, center of projections of the cameras, $\bar{c}_l$ and $\bar{c}_r$, and *epipoles*, $\bar{e}_l$ and $\bar{e}_r$ lie on the same plane. These observations lead to the definition of *epipolar constraint* that image points must satisfy. Given an image point in left image plane, $\bar{p}_l$, its correspondence in the other image plane, $\bar{p}_r$, lie on the line perpendicular to *epipolar plane* and passing through right *epipole*,

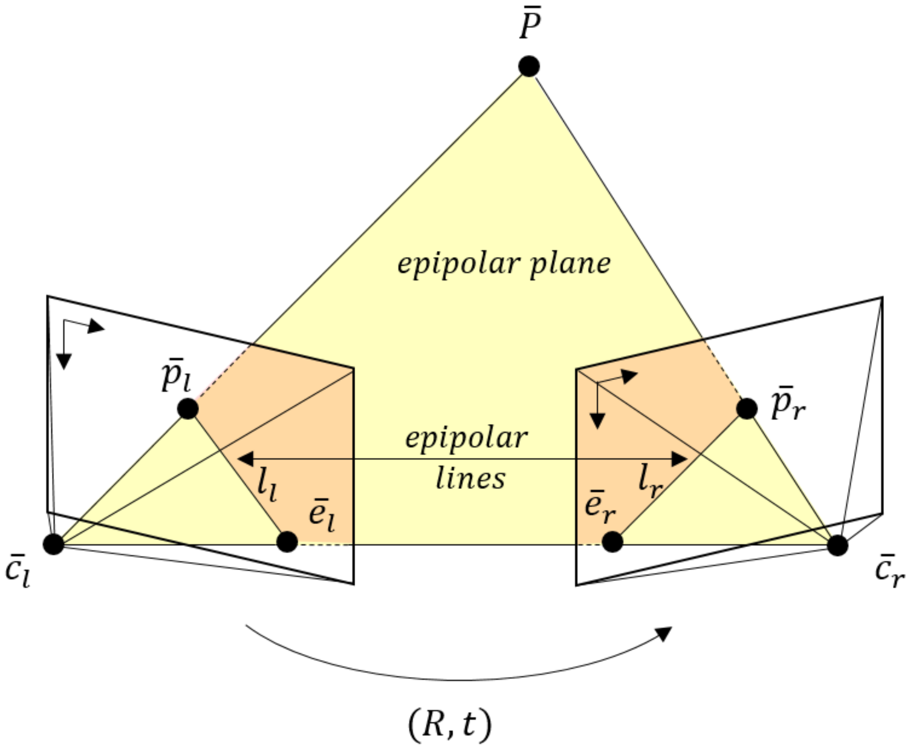


Figure 1.2: Illustration of epipolar geometry.

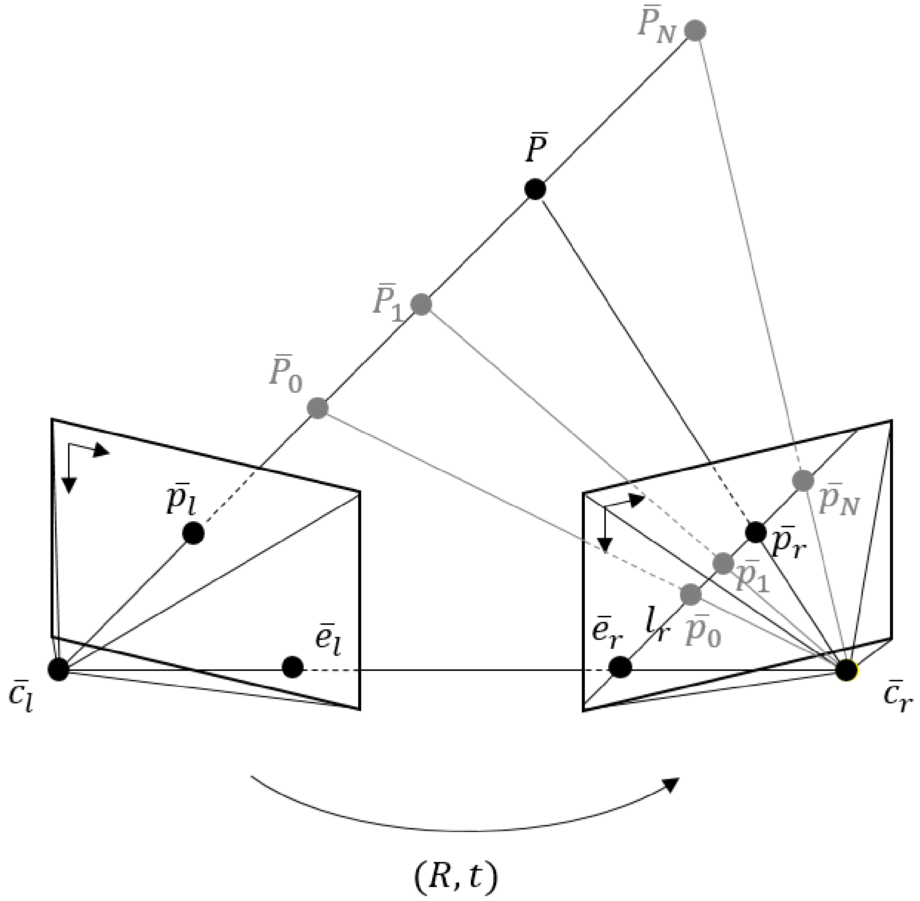


Figure 1.3: Epipolar constraints. The correspondence of the pixel, $\bar{p}_l$, in the left image should lie on the line l_r in the right image.

$\bar{e}_r$. This constraint restricts the search space to a line as shown in Fig. 1.3 and this line is called as *scanline.* With known relative rotation and translation, (R, t), of the left and right cameras, *epipoles* can be predetermined and for a given point in left image plane, $\bar{p}_l$, its corresponding *epipolar line* in right image plane, $\bar{l}_r$, can be determined using $\bar{c}_l$, $\bar{p}_l$ and *epipoles* so that correspondence search is performed for the match.

To simplify the depth extraction problem, the images from left and right cameras can be projected onto a common plane so that *epipolar lines* of two images coincide. In other words, one of the images is warped with respect to the other in order to make horizontal lines become *epipolar lines.* This process is called as *rectifying* the image pairs. Rectified image pairs can be considered as if they are captured from perfectly horizontally aligned two cameras image planes of which are coplanar. For this configuration of the cameras, there is a simple relation between depth information of a point and its projections onto image planes. This relation can be derived from Fig. 1.4 that illustrates the geometry of such two cameras. Coordinates of the scene point, $\bar{P}$, is relative to middle point of the line segment connecting the two centres of projections. $\bar{P}$ projects onto image planes at points $\bar{p}_l$, $\bar{p}_r$ and these points have coordinates $(x_l^{'}, y_l^{'})$, $(x_r^{'}, y_r^{'})$ respectively relative to centres of image planes. From *similarity* of triangles, the

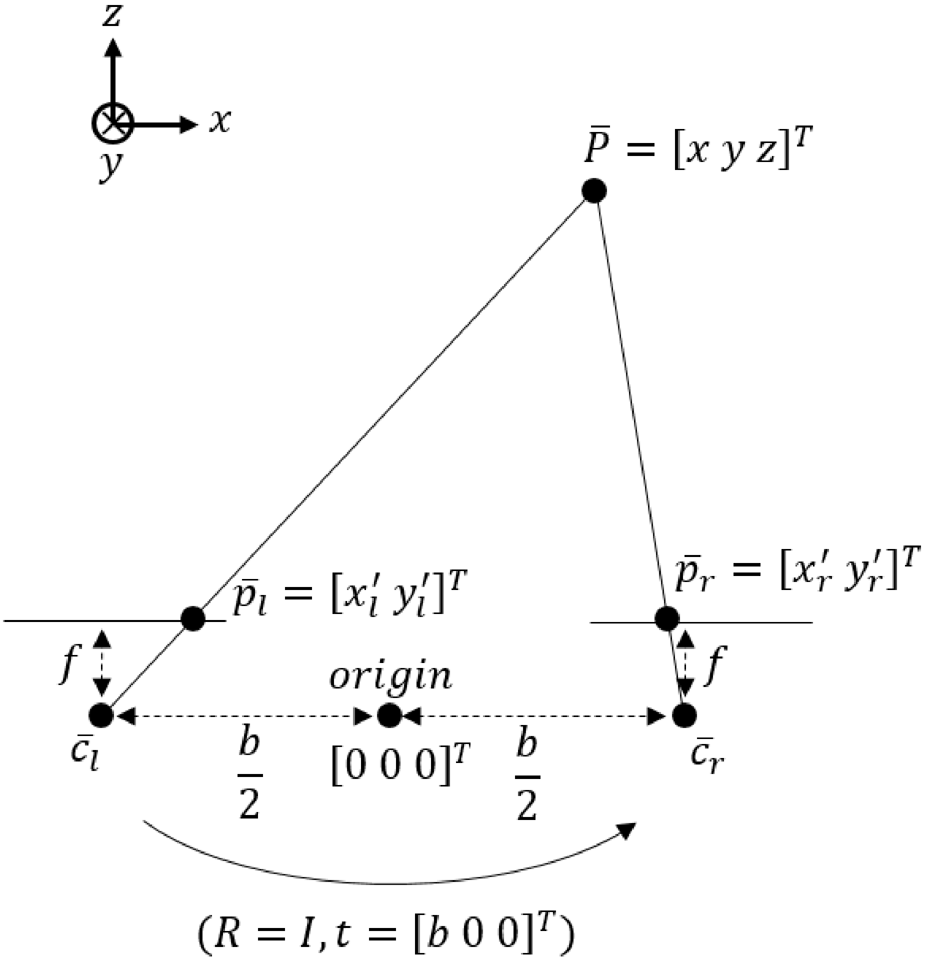


Figure 1.4: Simplest case of the stereo geometry. The two cameras are placed horizontally and their optical axes are parallel.

following is get:

$$\begin{gathered}\frac{x_l^{'}}{f} = \frac{x + \frac{b}{2}}{z} \; and \; \frac{-x_r^{'}}{f} = \frac{\frac{b}{2} - x}{z} \\ \frac{x_l^{'} - x_r^{'}}{f} = \frac{b}{z} \;\Rightarrow\; z = \frac{bf}{x_l^{'} - x_r^{'}} \; . \end{gathered} \tag{1.1}$$

The difference $x_l^{'} - x_r^{'}$ in Eq. 1.1 is defined as *disparity*, which can be considered as *inverse depth*. Eq. 1.1 shows that once disparity of a pixel is known, depth information of the point corresponding that pixel can be easily determined.

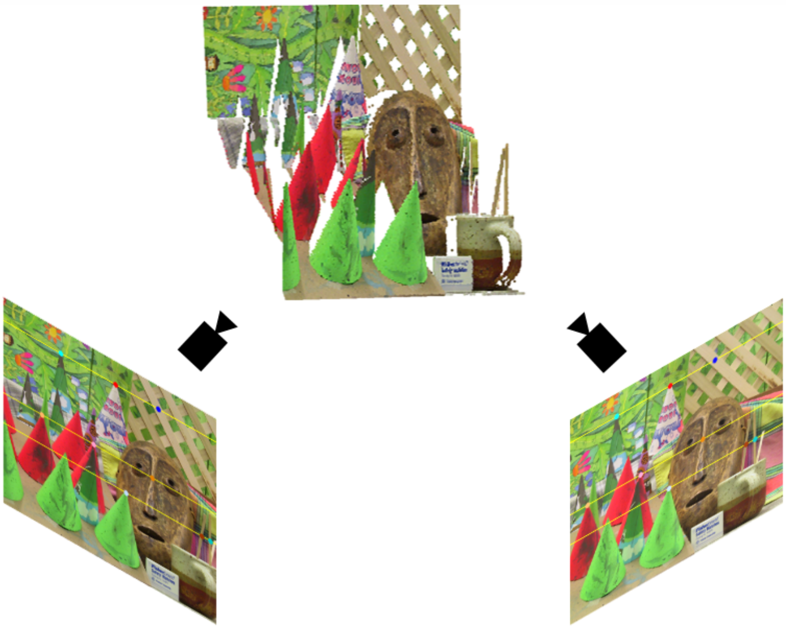


Figure 1.5: Illustration of the stereo matching problem on the *Cones* stereo pair from Middlebury evaluation data set [62]. The correspondences of the image points in one image should be found in the other image to extract depth information. The search is performed on epipolar lines indicated by yellow lines.

Finally, for rectified stereo image pairs, depth extraction problem can be defined as the problem of finding disparity of each pixel in the left or the right or both images by searching correspondences along the horizontal scanlines. This problem definition of depth extraction is valid only for the pixels corresponding scene points which are visible to both cameras. Moreover, computational complexity to estimate the disparities is an issue for applications demanding real-time processing. Therefore, there are additional problems for depth extraction except for the correspondence search problem. For the sake of completeness, these problems should also be defined.

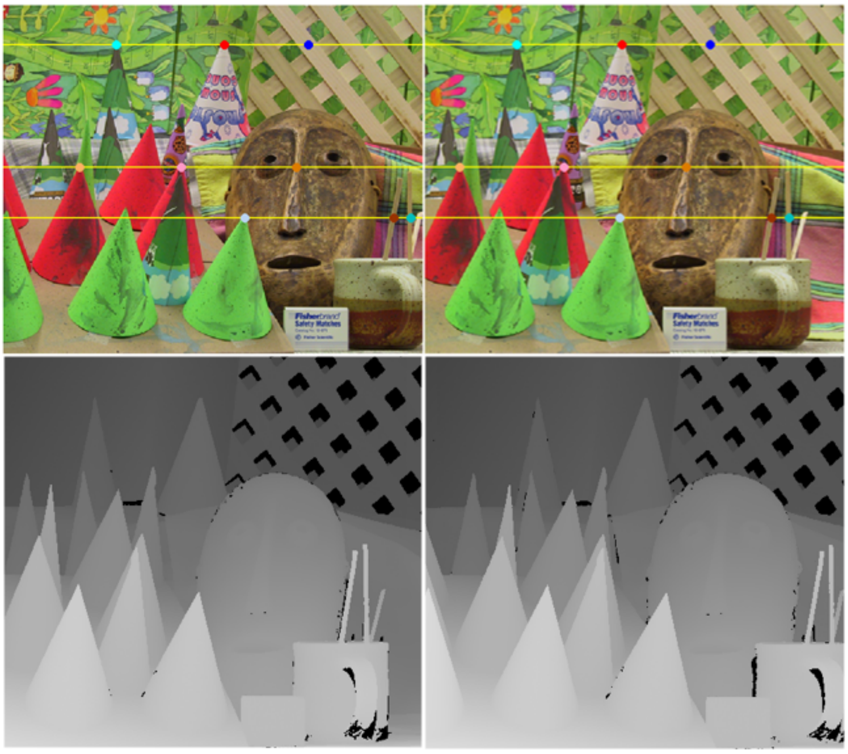
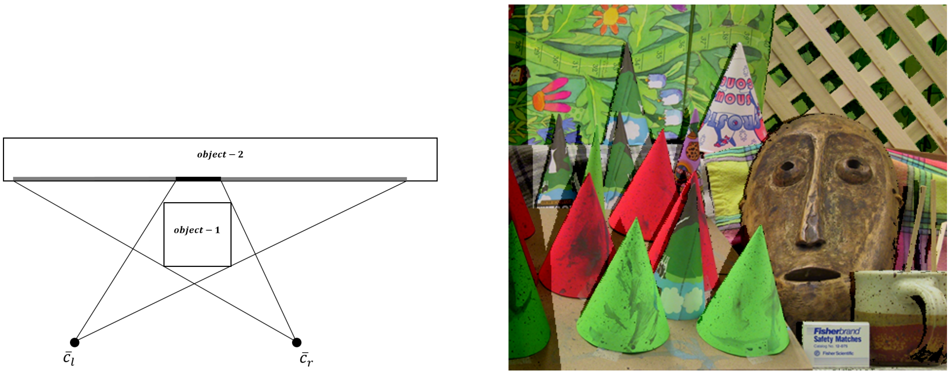


Figure 1.6: Stereo matching problem for rectified stereo image pairs. The problem becomes horizontal search for disparity. The disparity maps corresponding to the left and right images of *Cones* stereo pair [62] at the top are given in the bottom. Brighter regions are closer to the cameras.

1.1.1 Occlusion Handling

In stereo vision *occlusion* occurs when a point in the scene is prevented from being captured from one of the cameras or both of the cameras due to the objects in front of it. Occlusions can be classified as *full occlusion* and *half occlusion*. Former is the case when a point is not visible to both cameras and latter is the case when a point is visible to one camera but not the other. In Fig. 1.7b, a sample occlusion and half occluded regions of the stereo pair in Fig. 1.6 are illustrated. In depth extraction from stereo image pairs, only half occluded regions are considered. These regions are problematic regions not only because their disparity cannot be estimated via correspondence search but also they may lead to spurious matches of pixels causing erroneous estimation of disparities. Therefore, such regions should be excluded from correspondence search and disparity assignment should be handled differently. Consequently, *occlusion handling* problem can be defined as the problem of detecting occluded regions and predicting those regions disparities from disparities of visible regions.



(a) (b)

Figure 1.7: Occlusions in stereo images. (a) Formation of occlusion. Gray regions corresponds to half occluded regions. (b) Occluded regions of the left image of *Cones* stereo pair [62].

1.1.2 Computational Complexity

Computational complexity has no actual effect on disparity estimation. However, it should be considered as a problem when depth extraction must be performed within a limited amount of time. Therefore, algorithms with low computational costs are required for applications demanding fast processing of disparity estimation. Yet, it is known that there is a trade-off between disparity estimation performance and computational complexity, high performance comes with high computational complexity [1,62]. The challenge is to reduce the complexity while keeping the performance high. Hence, the problem due to computational complexity can be defined as the problem of developing an algorithm providing efficient disparity estimation with marginal decrease in performance, if any.

1.2 General Framework

Many algorithms have been developed to estimate dense disparity maps from rectified stereo image pairs so far [63]. These existing solutions for the disparity estimation problem can be investigated under a general framework. Algorithms take a stereo image pair as their input and treat one of the images as *reference image* and the other as *matching image.* According to the reference image, they produce a univalued function defined over image domain, $\Omega \in \mathbb{Z}^2$, where the reference image is defined over. This univalued function is commonly referred as *disparity map*, $D[x, y]$, which indicates the estimate of the amount of the offset to be add to the x coordinate of the pixel at location (x, y) in the reference image to determine the coordinate of its correspondence, $(x^{'}, y^{'})$ in the matching image, as,

$$x^{'} = x + sD[x, y], \;\; y^{'} = y \;, \tag{1.2}$$

where $s = -1$ when left image is the reference and right image is the matching image; $s = +1$ otherwise. Eq. 1.2 is inferred from Eq. 1.1. Two disparity maps for left and right images of a stereo pair are illustrated in Fig. 1.6. In order to obtain such disparity maps, two main algorithmic blocks is utilized in general: *matching* and *occlusion handling.* Matching step is the estimation of disparities of the pixels visible from both images, and the occlusion handling step is the estimation of disparities of the half occluded regions.

1.2.1 Matching

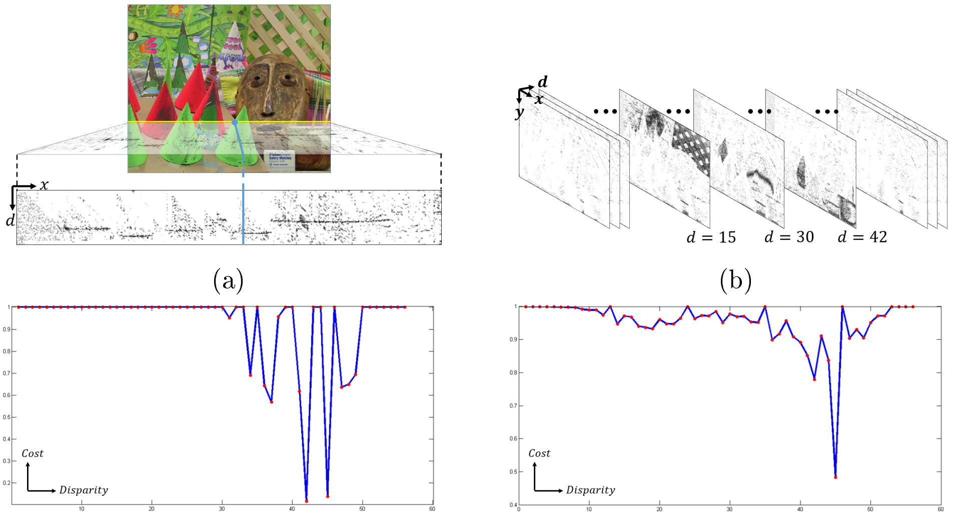
Matching is the process of finding correspondence of a pixel from the reference image in the matching image so that two pixels belong to the same scene point. To match pixels, several constraints are imposed in addition to the epipolar constraints. Three of these are quite general constraints and utilized in almost every algorithm. Firstly, *pixel intensity similarity constraint* is the most fundamental of these constraints, requiring that two pixel intensities in the reference and matching image must be almost equal if they belong to same scene point. Secondly, an important and necessary constraint is the *uniqueness constraint* which states there can be only one matching pixel in the matching image for a pixel in the reference image. This constraint is implicitly used to generate a univalued function, disparity map. In addition to the pixel matching, this constraint has a particular importance for occlusion detection, as explained in the following section. The last constraint is the *smoothness constraint.* This constraint presupposes that depths of physical objects in the real world vary smoothly and abrupt changes only occur at object boundaries; that is to say, disparity values in a local region should be similar except at the object boundaries. These constraints are utilized by the algorithms within an optimization framework in order to find matching pixels. Algorithms are mainly classified as *global*, *semi-global* and *local* according to optimization framework they use.

Regardless of which optimization framework they use, all algorithms work on the same domain, *disparity space*, which is the *Cartesian product* of image domain and disparity domain. Though disparity domain can be either continuous or discrete, usually it is discrete and integers from a finite domain in general. A point in the disparity space is represented by (x, y, d) and (x, y) coordinates always coincide with pixel coordinates of the reference image. After specification of the disparity space, an image is defined over the disparity space, representing the confidence or the costs of the matches implied by disparity map. This image is referred to as the *Disparity Space Image (DSI)* [11,86] or *cost volume* [35]. In Fig. 1.8, a DSI is visualized. A matching function, M, is used to determine the value of the DSI, $C[x, y, d]$, at each coordinate (x, y, d) as given by,

$$C[x, y, d] = M(I_r[x, y], I_m[x + sd, y]) \ , \tag{1.3}$$

where I_r and I_m represent vector valued reference and matching image functions, respectively. The entities of the vectors indicate the intensity values of different color channels. The matching function in Eq. 1.3 basically takes two intensity vectors and gives a measure of similarity between them. A matching function does not necessarily have to take two intensity vectors. It can take a block of images centred at related coordinates as well. Many pixel similarity/dissimilarity measures exist in the literature and a good evaluation of those can be found in [32].

DSI can be viewed as the realization of the similarity constraint as a cost function of a disparity assignment at a pixel location. Considering other constraints as well, algorithms try to assign optimal disparities to pixel locations according to this cost function mainly so that pixel matching is performed. At this point



(a)

(b)

(c)

(d)

Figure 1.8: Depiction of cost volume (DSI), its 2-D slices and 1-D cost function for each pixel. Darker regions correspond to lower costs. (a) A 2-D slice of the cost volume is illustrated. The slice corresponds to the yellow horizontal line on the image [62] and involves matching costs for all possible disparities for each pixel along the line. (b) 2-D slices of the cost volume under fixed disparity value, d, namely, the cost slice involves matching costs of the all pixels in the image for a particular disparity. (c) 1-D cost function for the particular pixel marked by blue dot on the line. The blue vertical line on the 2-D cost slice in (a) corresponds to 1-D cost function. (d) 1-D cost function after cost aggregation.

algorithms differ according to how they decide what an optimal assignment is.

Global methods [12,25] try to find a surface embedded in the cost volume/DSI with some optimality property. They try to *optimize total cost of a disparity map subject to smoothness constraints*. This aim is achieved by optimizing the following cost function, E,

$$\begin{aligned} E(D) &= E_{data}(D) + \lambda \cdot E_{smooth}(D) \\ E_{data}(D) &= \sum_{(x,y)\in\Omega_r} C[x,y,D[x,y]] \ , \end{aligned} \tag{1.4}$$

where Ω_r is the domain of reference image function. The smoothness constraint is introduced as a penalty term in the energy function defined in Eq. 1.4 and λ is the parameter controlling influence of this penalty. Common approach is to solve this optimization problem is proper *Markov Random Field* (MRF) modeling [44]; since, there are efficient optimization methods [12, 25] that can be utilized for MRF based problems .

Semi-global methods [6, 31, 77] deal with scanlines and optimize a cost function similar to the one in Eq. 1.4 for each scanline individually. This can be viewed as finding the path on the cost surface between two ends of a scanline that has the optimal cost. In Fig. 1.8a, the related slice of the cost volume for a scanline is shown. This cost slice is the surface where the optimal path is to be found. *Dynamic Programming* (DP) [5] is usually utilized for the solution of these scanline optimization problems.

Local methods [26,34,38,87] treat each disparity candidate independently and do not define a combined cost function to optimize. Instead, they create a cost function for each pixel and perform disparity assignment separately corresponding to an optimal cost for each one. Simplest way to create a cost function for a pixel, $C_{(x,y)}[d]$, is to get the matching costs of all possible disparity assignments for that pixel. This corresponds to the curve in the cost volume along the disparity dimension under fixed image coordinates; that is, $C_{(x,y)}[d] = C[x, y, d]$. Two examples of a 1-D cost function for a pixel are shown in Fig. 1.8c and 1.8d. Generally, optimization of a 1-D function directly extracted from the cost volume does not yield satisfactory results. This is mainly lacking of similarity constraint alone. Therefore, general approach is performing *cost aggregation* prior to extract 1-D cost function from the cost volume. Cost aggregation is basically replacing the matching costs of the pixels with the sum or average of the matching costs of the pixels in a local region. In other words, an aggregated cost of a pixel for a particular disparity, $C_f[x, y, d]$, is given as,

$$\begin{aligned} C_f[x,y,d] = \sum_{(x',y')\in\mathcal{N}_{(x,y)}} & w(x,y,x',y')C[x',y',d] \\ \sum_{(x',y')\in\mathcal{N}_{(x,y)}} & w(x,y,x',y') = 1 \end{aligned} \tag{1.5}$$

where $\mathcal{N}_{(x,y)}$ is a local region associated with (x, y) and w is some weighting coefficient. The cost aggregation in Eq. 1.5 indicates 2-D weighted average filtering on each slice of the cost volume as a consequence of a given disparity value (Fig. 1.8b). After generation of proper cost functions, the disparities are assigned *winner take all* (WTA) optimization scheme for each pixel individually as,

$$D[x, y] = \arg\min_{d} C_{(x,y)}[d] \ . \tag{1.6}$$

Disparity assignment according to the aggregated costs implies that the optimality of a disparity assignment for a pixel depends on the optimality of that assignment for the pixels in the local region.

1.2.2 Occlusion Handling

Disparity estimation problem for occluded regions is dealt with mainly in three ways. First one is implicit handling in which the problem is left to be solved by the effect of smoothness constraint [12, 39, 41, 71]. The other two are explicit handling during matching step and leaving the problem to a post processing step. Occlusion handling during matching step is encountered in global [42, 80, 84] or semi-global [4, 6, 11, 28, 77] optimization based methods. In these methods, an occlusion cost term is introduced to the cost function in Eq. 1.4. This is obtained by either embedding the occlusion cost in data term [80, 84] or explicitly defining a cost term related to occlusion [4, 6, 11, 28, 42, 77]. Leaving occlusion handling problem as a post processing step is a general approach and it is mostly preferred in local methods. The post processing consists of two main steps which are occlusion detection and filling. Occlusion detection can be done either by using disparity map of a single image or by using disparity maps of the left and right images together. The former is referred as *asymmetric* [2, 67] and the latter is referred as *symmetric* [15, 51] occlusion handling. The most popular approach for symmetric occlusion handling is *left-right consistency checking* [15].

1.2.3 Compexity Reduction

The computational complexity of the disparity estimation algorithms is mainly due to the matching step. According to the type of the algorithm, the source and degree of the complexity differ.

Optimization of the cost defined in 1.4 is NP-hard problem in general for global algorithms. Efficient algorithms [12, 25] reduce this to polynomial complexities which is proportional to image size, disparity search range and number of iterations. Semi-global methods perform optimization in each scanline independently by DP. Hence, they also have polynomial complexities, $\mathcal{O}(N \cdot D^2)$, with the image size, N, and square of the disparity search range, D. A common approach to reduce the complexity further is to reduce the search range for disparities. For this purpose, *coars-to-fine* (CTF) approaches are utilized [77, 85].

Complexity of the local methods are mainly due to cost aggregation step and the disparity search range. As stated in Sec. 1.2.1, a cost aggregation is performed over a local region of each pixel. The size of this local region is determined by the filter kernel and is a factor of complexity. Therefore, complexities of local methods are $\mathcal{O}(N \cdot W \cdot D)$ in general, where N, W and D are the image size, filter kernel size and disparity search range, respectively. Among the methods introduced so far, some of the techniques reduce the complexity due to the disparity range by using hierarchical [36, 37, 66] or stochastic approaches [9, 50], whereas some others decrease the complexity due to the kernel size by exploiting the regular structure of cost volumes [17, 35, 60]. The methods exploiting this regularity make cost aggregation step independent of filter kernel size by getting rid of repetitive calculations via systematic operations. The main challenge is to reduce the complexity due to both the disparity range and the kernel size jointly. Irregular data access scheme of CTF [36, 37, 66] or stochastic methods [9, 50] is highly opposed to regular and systematic computing style of the filtering methods that can reduce the complexity due to the kernel size [17, 35, 60]. Therefore, the approaches to reduce the complexity due to the disparity range introduces kernel size dependent complexity that cannot be reduced by the efficient filtering methods available in the literature [17, 35, 60].

1.3 Scope of the Thesis

In this thesis study, the advances in the local stereo matching methods are followed and the problem of reducing the complexity due to the disparity search range and the filter kernel size is attacked.

Existing solutions in the literature for complexity reduction during the cost aggregation step are reviewed. The reasons of the inapplicability of these solutions to the methods aiming complexity reduction on disparity search range are investigated. Considering those reasons, a novel recursive edge-aware filtering based cost aggregation procedure is proposed to perform an efficient cost aggregation to be used in the methods aiming complexity reduction on disparity search range. In this way, the complexities due to both sources are reduced.

From the perspective of complexity reduction on disparity search range, a coarse-to-fine approach is utilized. The reasons for the performance drops in CTF approaches are analysed. A novel method for disparity information transfer across the scales is proposed to improve the disparity estimation performance by preventing error propagation across the scales. Moreover, the occlusion handling problem is coupled with the disparity information transfer problem and solved concurrently.

Finally, by combining the two proposed main blocks, an efficient stereo matching method is presented and it is theoretically and experimentally shown that the complexity of the method is linear with the image size.

1.4 Outline of the Thesis

This thesis consists of four chapters. Following the introduction to depth estimation problem from stereo image pairs with some preliminaries, general solution methods are summarized in Chapter 1.

Chapter 2 is devoted to the literature review of the stereo matching algorithms by expanding the methods presented in Chapter 1. Algorithms are classified into groups and key points of each group are detailed. The local stereo matching methods are emphasized more and the complexity reduction approaches of local stereo methods are reviewed in detail, since the prior art that is most related with the work presented in this study involves the methods aiming complexity reduction. As a final section, the occlusion handling methods are classified and mentioned.

Chapter 3 presents the proposed approach consisting of two main algorithmic blocks for the complexity reduction problem in stereo matching. The proposed approach is analysed by the results of the experiments and compared with the state-of-art methods. It is theoretically shown that the proposed approach has a complexity that is independent of the disparity search range and the cost aggregation step. This linear time complexity with the image size is also validated by experiments.

Finally, Chapter 4 summarizes the proposed method for efficient stereo matching and draws conclusions from the experiments. Additionally, future directions for possible applications of the presented method to existing problems are also presented.

CHAPTER 2

RELATED WORK

In this chapter, a literature review on stereo matching methods is presented. During the last decades many algorithms have been developed for stereo correspondence problem and a good taxonomy for these algorithms is available in [13, 63]. The desired depth maps are expected to mostly reduce the observed pixel intensity cost, while being smooth and discontinuity preserving. In Sec. 1.2, it is seen that stereo matching algorithms generally consist of three main steps: matching cost calculation [32], disparity optimization and post processing to refine disparity estimation. The algorithms are classified as global, semi-global and local according to handling of optimization step. In addition to this classification, cooperative methods that fuse local and global optimization can be added as an additional class. In the following sections, these classes of algorithms are presented broadly and the key points which lead these algorithms to the state-of-art are explained in detail. The prior art that is most related with this thesis study includes studies on complexity reduction in local algorithms. Therefore, local methods are also reviewed according to the complexity reduction approaches. As a final section, occlusion handling methods utilized by stereo matching algorithms are given with classification.

2.1 Global Methods

Global methods [7, 10, 12, 18, 25, 33, 39, 41, 42, 45, 68, 80, 84, 85, 89, 93] estimate the disparities of all of the pixels at once by optimizing a global cost function. In general, the objective can be considered as the minimization of the cost function.

The cost function generally consists of two terms: data term and smoothness term. Data term is related to the cost volume which holds correspondence costs of the pixels at given disparities. The smoothness term is a penalty term to include the smoothness constraint in the optimization process. The aim of the smoothness term is to enforce pixels to have similar disparities with their neighbours. General cost function, E, for a disparity map, D, is,

$$\begin{aligned} E(D) &= E_{data}(D) + \lambda E_{smooth}(D) \\ E_{data}(D) &= \sum_{(x,y)\in\Omega_r} C[x,y,D[x,y]] \\ E_{smooth}(D) &= \sum_{(x,y)\in\Omega_r} e_{smooth}(x,y) \end{aligned} \tag{2.1}$$

where Ω_r is the domain of reference image function, C is the cost volume and λ is the parameter controlling the degree of influence of this penalty. Smoothness term for a pixel, e_{smooth}, is often picked as a function depending on disparity difference and image intensity difference of neighbouring pixels. The reasoning behind the dependency on intensity difference is to adapt the penalty cost to intensity edges where disparity discontinuities generally occur. Penalty term for a pixel at (x,y) with a local neighbourhood $\mathcal{N}_{(x,y)}$ can be expressed as,

$$e_{smooth}(x,y) = \sum_{(x^{'},y^{'})\in\mathcal{N}_{(x,y)}} \rho(D[x,y], D[x^{'},y^{'}], I_r[x,y], I_r[x^{'},y^{'}])\ . \tag{2.2}$$

In general, the function ρ is factorized as:

$$\rho(...) = \rho_i(I_r[x,y], I_r[x^{'},y^{'}])\rho_d(D[x,y], D[x^{'},y^{'}]) \tag{2.3}$$

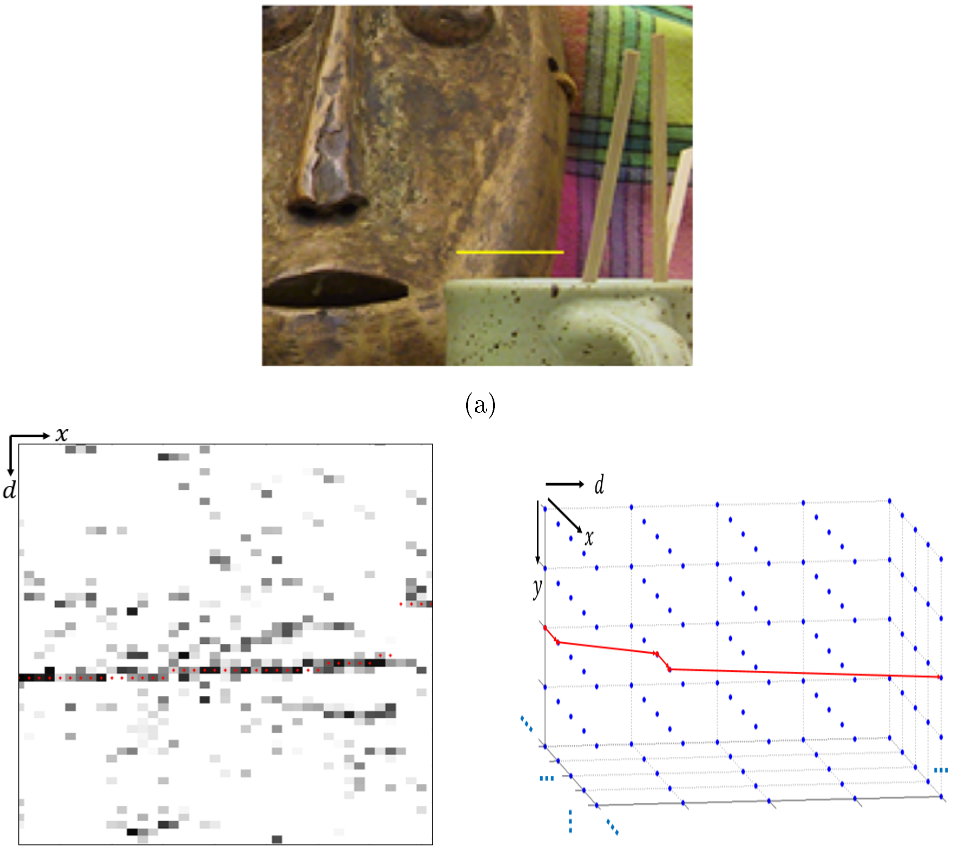
where ρ_i takes value inversely proportional to color intensity difference between two pixel in the reference image, I_r, and ρ_d is often monotonic increasing function of disparity differences of two pixels. The function ρ for a pixel p at location (x,y) and a pixel q at location $(x^{'},y^{'})$ is generally referred as $\rho_{p,q}(D[p],D[q])$.

The minimization of the cost function in Eq. 2.1 is an NP-hard problem in general. However, this optimization problem can be solved efficiently by proper *Markov Random Field* (MRF) modeling [44] for which efficient optimization methods exist. According to the survey conducted in [70], the state-of-art methods are based on *graph cuts* (GC) [12] and *belief propagation* (BP) [25]. The methods that exploit GC or BP perform MRF modeling on regular image grid and minimize the cost function iteratively. These methods provide accurate disparity estimations even in the presence of noise or textureless regions that are common problems for performance drops of stereo matching algorithms. However, pixel-wise approach may still yield erroneous estimations at object boundaries. To overcome this problem, surface fitting [10, 45, 68, 84] and region based MRF modeling are proposed in several studies [7, 18, 33, 41, 93]. In both approaches, an initial over segmentation of the reference image according to color-wise similarity is performed in order to form pixel groups of similar color. In surface fitting methods, each group of pixels are enforced to belong to the same surface. In [84], this aim is achieved by an iterative process. First, an initial disparity map is estimated and the surface fitting is performed for region of each group from previously estimated disparity map. According to the new disparities estimated by surface fitting, the data term of the cost function is updated in such a way that the disparity assignments different from the result of the surface fitting are penalized. The iteration is completed by minimizing the cost function by using this new data term. In [10, 45, 68], surface fitting is fused with the optimization scheme by adding a cost term forcing disparities of color-wise similar pixels to lie on the same surface.

In region based methods [18, 33, 41, 93], instead of using single pixels as nodes, pixel groups of similar color are used and optimization strategy is performed on the graph constructed by those groups. In this way, performance of global methods is increased at object boundaries and textureless regions. Yet, this increase in performance comes with increase in computational complexity; since, violation of regular grid structure of the MRF network avoids fast processing of message passing in BP based methods.

2.2 Semi-global Methods

Semi-global methods, in general, perform a global optimization in each scanline independently and the term semi-global comes from this strategy. Semi-global methods [4, 6, 11, 28, 29, 31, 43, 76, 77] minimize a cost function by using *dynamic* *programming* (DP) [5]. DP is a mathematical method which decomposes optimization problems into simpler sub-problems so that the complexity is reduced. In semi-global methods, a cost function is minimized for each scanline independently. This process can be viewed as finding the minimum cost path on the slice of the cost volume from leftmost column to the rightmost column. The slice corresponds to the 2-D cost image extracted from the cost volume for the scanline. Fig. 2.1 depicts a part of a cost slice and the minimum cost path on that slice is shown by red dots. In [4,6,11,28,77], a 1-D cost function is defined



(a)

(b)

(c)

Figure 2.1: Illustration of minimum cost path to be searched by semi-global based methods. (a) A part of the reference image. The disparities on the yellow line is considered. (b) The 2-D matching cost slice corresponding to the yellow line in (a). Red dots represents the minimum cost path obeying particular constraints (the brighter the pixel, the more cost it has). (c) A simple illustration of the minimum cost path on the disparity space, namely, the domain of the cost volume.

for scanlines and this function is generally in the form,

$$E(D_s) = \sum_{(x,y)\in s} C[x,y,D_s(x,y)] + \sum_{(x,y)\in s} e_{occl}(x,y) \ . \qquad (2.4)$$

In Eq. 2.4, D_s is the disparity map of a scanline s and e_{occl} is the penalty term when a pixel is considered to be occluded. The methods [4,6,11,28,77] assume that there is an ordering constraint between neighbouring pixels along the same scanline as solving the problem. In this way, the search space is reduced and occlusions can be handled. In [29,31,43,76], a global cost function like the one introduced in Eq. 2.1 is defined for the entire image and this cost function is minimized for each scanline independently by using DP as,

$$e(x,y,d) = C[x,y,d] + \min_k \left\{ e(x-1,y,k) + \rho(d,k,I_r[x,y],I_r[x-1,y]) \right\}, \quad (2.5)$$

where ρ is presented in Eq. 2.2. $e(x,y,d)$ holds the minimum cost up to pixel (x,y) when its disparity is d. It can be observed that smoothness constraints is considered only for adjacent pixels in scanlines. After the calculation of e for all pixels at all disparities, the disparity of the rightmost pixel in the scanline is determined by selecting the disparity assignment with the minimum cost. Then this information is propagated to the adjacent pixel on the left so that that the disparity of that pixel is determined. This process is repeated until leftmost pixel in the scanline is reached.

Semi-global methods reduce the complexity of the optimization procedure to a polynomial complexity, $\mathcal{O}(N \cdot D^2)$, with the image size, N, and the square of the disparity search range, D. This is the main advantage of semi-global algorithms; since, it allows fast processing. However, the results are often poor with streaking artifacts due to absence of inter-scanline consistency constraints. To overcome this problem, DP methods that also consider consistency between scanlines are proposed [29,31,43,76,77]. In [29,77], the consistency is exploited via vertical support of cost function and post-processing to reduce streaking artifacts. In [31], smoothness among neighbouring pixels is enforced by fusion of the DP framework with a cost aggregation concept. It is performed in two steps. First one is performing DP not only in horizontal scanlines, but also in different 1-D directions, such as vertical and diagonal. Then, a 1-D cost for each pixel is formed for each disparity candidate after averaging the costs from these multiple directions and the disparities are assigned according to WTA minimization of these 1-D cost functions. In [43, 76], a minimum spanning tree is extracted from the image and DP is performed in that tree so that smoothness constraint is expanded to entire image. In [43], a region based approach is followed for further improvement and the tree is extracted from the over segmented image.

2.3 Local Methods

Unlike global and semi-global methods, local methods do not define a global cost function to optimize. Instead, they create a cost function for each pixel by treating each disparity candidate independently and perform disparity assignment using WTA optimization as given by Eq. 1.6. A cost function, $C_{\bar{x}}[d]$, for each pixel at location $\bar{x} = (x, y)$ can be inherited directly from the cost volume, $C[\bar{x}, d]$, constructed using pixel-wise matching costs, as,

$$C_{\bar{x}}[d] = C[\bar{x}, d] \ . \tag{2.6}$$

However, in that case, smoothness is not enforced among pixels and this results in spurious matches due to weak distinctiveness and noise susceptibility of pixel-wise matching cost alone. Therefore, the cost of a disparity assignment to a pixel is determined by aggregating the matching costs of the pixels for that disparity assignment through a summation or an averaging over a local region[1]. In this way, optimality of a disparity assignment for a pixel depends on the optimality of that assignment for the pixels in the local region. The cost aggregation via summation can be given as,

$$\tilde{C}[\bar{x}, d] = \sum_{\bar{x}' \in \mathcal{N}_{\bar{x}}} C[\bar{x}', d] \ , \tag{2.7}$$

and the cost aggregation via averaging is,

$$\tilde{C}[\bar{x}, d] = \sum_{\bar{x}' \in \mathcal{N}_{\bar{x}}} w_{\bar{x},\bar{x}'} C[\bar{x}', d] \ , \ \sum_{\bar{x}' \in \mathcal{N}_{\bar{x}}} w_{\bar{x},\bar{x}'} = 1 \ , \tag{2.8}$$

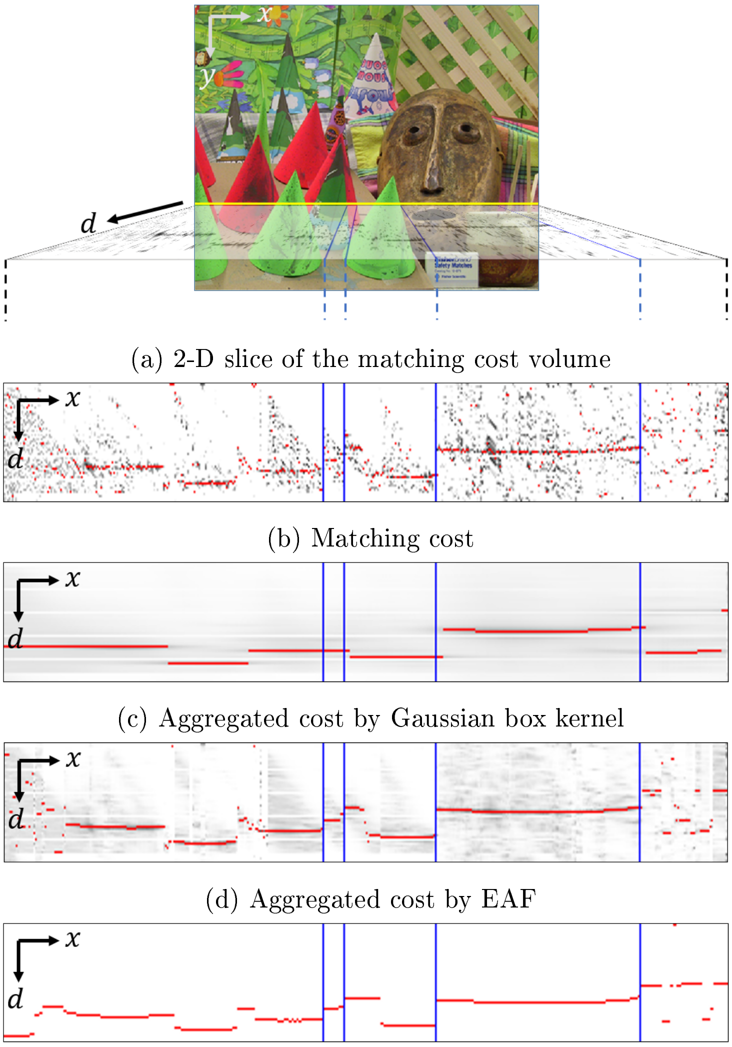
[1] Instead of performing cost aggregation, more robust and distinctive matching cost functions can be utilized as well [88]. The analysis of those strategies are kept out of the scope of this study.

where $\mathcal{N}_{\bar{x}}$ represents a local region of the pixel of interest, $\bar{x}$, and $w_{\bar{x},\bar{x}'}$ is the weight of a pixel in the local region. The local region is generally a window and there is an implicit smoothness assumption within the window. All the pixels inside the window are enforced to have the same disparity value. Cost aggregations presented in Eqs. 2.7 and 2.8 can be viewed as filtering the 2-D cost slice of the matching cost volume when a disparity is fixed. Therefore, for the rest of the section the terms *filtering* and *cost aggregation* can be used interchangeably.

Local methods are simpler compared to other methods and they do not require any iteration step. For this reason, local methods are preferred when fast processing is desired. The filtering process presented in Eq. 2.7 is actually well-known box filtering [78] and can be performed in a constant time independent of the window size via utilization of integral images [20]. However, simplicity and fast processing come with a prize which is the degradation of performance. Especially at the regions where depth discontinuities are present, the smoothness assumption within a window is no more valid at all due to the pixels from both foreground and background in the window. This causes foreground/edge fattening effect in the estimated disparity maps, yielding poor estimations at object boundaries. Moreover, the smoothness assumption is violated also for slanted surfaces. Yet, weighted averaging reduces sensitivity of the filtering to those regions. In Fig. 2.2c, the cost slice corresponding to a horizontal line after filtering the cost volume with a spatial Gaussian kernel is presented and foreground fattening effect can be observed from the filtered cost slice by examining the minimum points shown by red.

Early researchers have concentrated on edge fattening problem [26, 38] and developed methods to reduce sensitivity to the object boundaries via multiple windowing or adapting the window size and location. However, early attempts have been inferior compared to their global counterparts [62] until the introduction of edge-aware filtering (EAF) in stereo matching [74]. In [74], researchers apply **joint**[2] ***bilateral filtering*** [87] for cost aggregation. Bilateral filter is a

[2] Joint filtering is the filtering of a signal by using information of another signal. In stereo matching, cost volumes are filtered using image functions.



(a) 2-D slice of the matching cost volume

(b) Matching cost

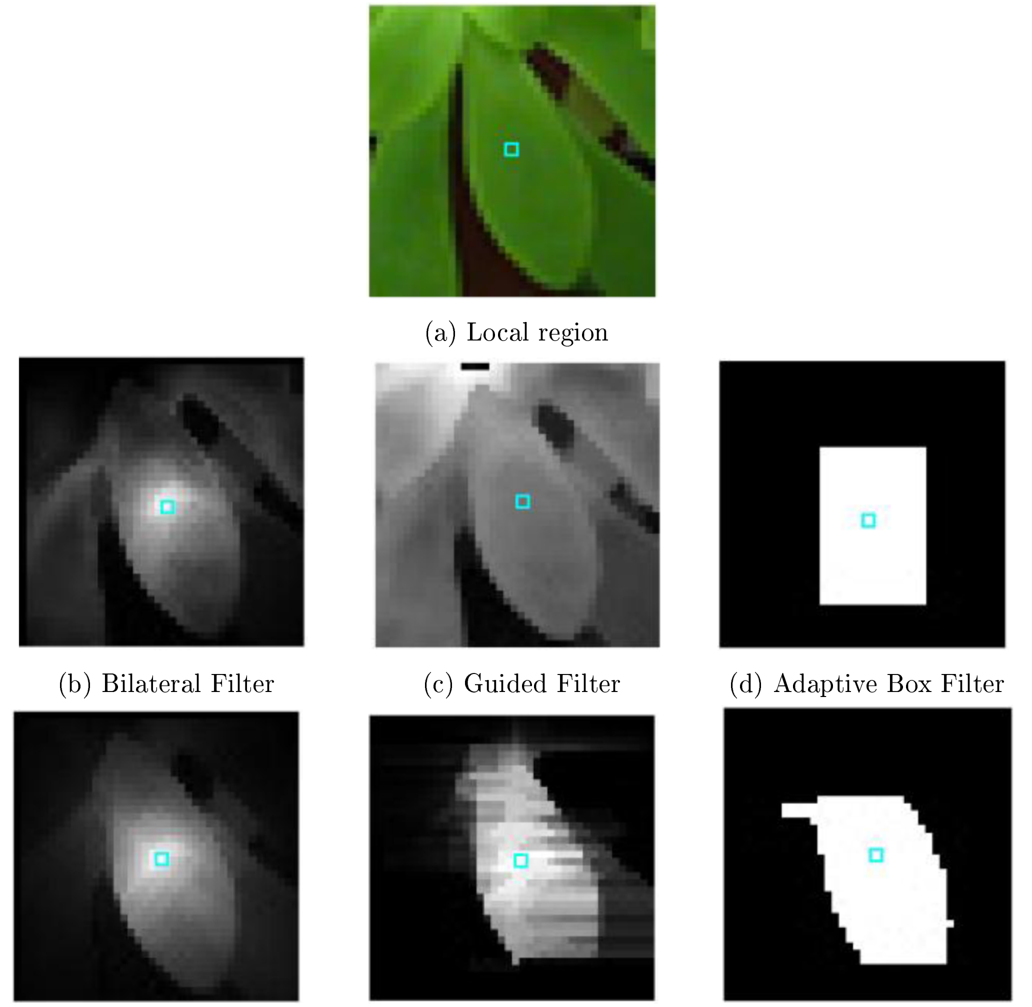
(c) Aggregated cost by Gaussian box kernel

(d) Aggregated cost by EAF

(e) Ground truth disparity maps

Figure 2.2: Comparison of the disparity estimations according to different cost aggregation methods. Red dots indicate the minimum points of the 1-D cost functions of each pixel on the horizontal line. Vertical blue lines correspond to some discontinuities in the reference image. (a) The reference image [62] and the 2-D cost slice corresponding to the yellow horizontal line. Cost slice is extracted from matching cost volume where darker regions correspond to lower costs. (b) Disparity estimation using WTA optimization without cost aggregation. (c) Disparity estimation after performing Gaussian box filtering of the cost volume. (d) Disparity estimation after EAF. The effect of cost aggregation to prevent spurious matches and the effect of EAF can be observed by comparing (b)-(c),(d) and (c)-(d), respectively.

weighted averaging filter. The weights of the pixels within the window are determined according to spatial and color distances to the pixel to be filtered. The



(a) Local region

(b) Bilateral Filter

(c) Guided Filter

(d) Adaptive Box Filter

(e) Geodesic Support Filter

(f) Recursive EAF

(g) Arbitrary Shaped Cross

Figure 2.3: Comparison of the resultant support regions of the different methods for the pixel marked by blue square. Support regions are retrieved from [16].

cost aggregation in this case is,

$$C_f[\bar{x}, d] = \frac{\sum\limits_{\bar{x}' \in \mathcal{N}_{\bar{x}}} F_S(\bar{x}, \bar{x}') F_R(I_r[\bar{x}], I_r[\bar{x}']) C[\bar{x}', d]}{\sum\limits_{\bar{x}' \in \mathcal{N}_{\bar{x}}} F_S(\bar{x}, \bar{x}') F_R(I_r[\bar{x}], I_r[\bar{x}'])} . \tag{2.9}$$

If the same convention in Eq. 2.8 is used then the filter weights become,

$$w_{\bar{x},\bar{x}'} = \frac{F_S(\bar{x}, \bar{x}') F_R(I_r[\bar{x}], I_r[\bar{x}'])}{\sum\limits_{\bar{y} \in \mathcal{N}_{\bar{x}}} F_S(\bar{x}, \bar{y}) F_R(I_r[\bar{x}], I_r[\bar{y}])} \tag{2.10}$$

where F_S is the spatial weight function to assign weights according to spatial distance between pixel locations and F_R is the range weight function to assign weights according to color intensity differences of the pixels obtained from reference image function I_r. Although many functions are possible, Gaussian functions are used in general, as,

$$
\begin{aligned}
F_S(\bar{x},\bar{x}^{'}) &= \mathrm{e}^{-\frac{1}{2}\left(\frac{\| \bar{x}-\bar{x}^{'} \|_2}{\sigma_s}\right)^2} \\
F_R(I_r[\bar{x}],I_r[\bar{x}^{'}]) &= \mathrm{e}^{-\frac{1}{2}\left(\frac{\| I_r[\bar{x}]-I_r[\bar{x}^{'}] \|_2}{\sigma_r}\right)^2},
\end{aligned}
\tag{2.11}
$$

where σ_s and σ_r are decay control parameters. Bilateral filter provides color adaptive averaging which enforces smoothness only for color-wise similar pixels in a local region and preserves depth discontinuities at object boundaries. One drawback of the bilateral filter is weak consideration of connectedness among the support pixels. This situation is depicted in Fig. 2.3. It is not always true that all color-wise similar pixels within a support window belong to the same surface. They might come from different objects and consequently have different disparities. Assignment of high weights to all color-wise similar pixels may yield erroneous disparity estimations. Hence, spatial weighting function of the bilateral filter deals with this problem to some degree by assigning lower weights to distant pixels. However, in some cases, especially in the presence of weak texture, contributions of the distant pixels become important for reliable estimations and penalizing them due to distance prevents this benefit. ***Geodesic support filter*** is proposed to overcome this problem [34]. Geodesic support filter assigns weights to the support pixels according to geodesic distances [72] to the pixel to be filtered. Geodesic distance between two pixels, $GD(\bar{x},\bar{y})$, is the cost of the minimum cost path between these two pixels when color dissimilarities are considered as costs. The geodesic distance can be defined more explicitly, if an image is considered as a graph that has the pixels as its nodes and the edges of it are formed by connecting each pixel to its immediate 8-neighbours surrounding it. A node can reach another node via edges in this graph. A path, γ, linking two pixels can now be defined as an ordered set of edges. It can be also considered as an ordered set of nodes to be passed. If there are n nodes, $\{\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_n\}$, on the path linking pixel node $\bar{x}$ to pixel node $\bar{y}$ then the cost of this path, $c(\gamma)$, becomes,

$$c(\gamma) = \sum_{i=1}^{n-1} \parallel I_r[\bar{x}_i] - I_r[\bar{x}_{i+1}] \parallel_2, \ \bar{x}_1 = \bar{x}, \ \bar{x}_n = \bar{y}, \tag{2.12}$$

and if $\Psi(\bar{x}, \bar{y})$ is the set of all paths linking pixel $\bar{x}$ to pixel $\bar{y}$ then the geodesic distance, $GD(\bar{x}, \bar{y})$, can be given as

$$GD(\bar{x}, \bar{y}) = \min_{\gamma \in \Psi(\bar{x}, \bar{y})} \{c(\gamma)\} \ . \tag{2.13}$$

Geodesic distance of two color-wise similar pixels from different objects is supposed to be high provided that an edge is present at the object boundaries. In Fig. 2.3, a comparison of the weight assignments of the bilateral and geodesic support filter is provided. It can be observed that geodesic distance based weight assignment is consistent with connectedness of the support pixels. The assessment of geodesic distance based weights has high computational load. Two main approaches are exploited to determine the geodesic distances [64, 72]. In [64], *Fast Marching Method*, which is a *wave-front* algorithm, is proposed. Fast Marching Method is similar to well-known Dijkstra's algorithm in the manner of iterative propagations from the center pixels of the support windows. In [72], with the spirit of dynamic programming, researchers propose a *raster-scan* approach which is scanning the image by starting from top-left, ending at bottom-right. In this method, forward and backward passes are performed iteratively until convergence to determine geodesic distances. Generally, few passes are formed to approximate the distances in order to save computation. To compute geodesic distances within a window to the center pixel, $\bar{x}_c$, via this method, the center pixel's geodesic distance is set to 0 and the geodesic distance of the rest is set to some large value. Then, each distance is updated as,

$$GD(\bar{x}, \bar{x}_c) = \min_{\bar{y} \in K_{\bar{x}}} \{|I_r[\bar{x}] - I_r[\bar{y}]| + GD(\bar{y}, \bar{x}_c)\} \ , \tag{2.14}$$

where $K_{\bar{x}}$ is the set of neighbouring pixels of $\bar{x}$ that are indicated by the kernels given in Fig. 2.3. After the geodesic distances are determined via some method, the filtering presented in Eq. 2.9 is performed by replacing spatial and color based weight functions with the weight function

$$F_G(\bar{x}, \bar{x}') = \mathrm{e}^{-\frac{1}{2}\left(\frac{GD(\bar{x}, \bar{x}')}{\sigma_g}\right)^2} \ . \tag{2.15}$$

Apart from the approaches that are based on geodesic support filtering, there are iterative approaches exploiting the geodesic distances implicitly [22, 57]. These approaches mimic the geodesic diffusion process and perform cost aggregation iteratively by passing the image in vertical and horizontal directions. At each pass, the cost information of the pixels are propagated according to color similarities. Consequently, the pixels are supported by the pixels that are close in the geodesic sense. When compared to geodesic support filtering based methods, these methods have better computational complexity.

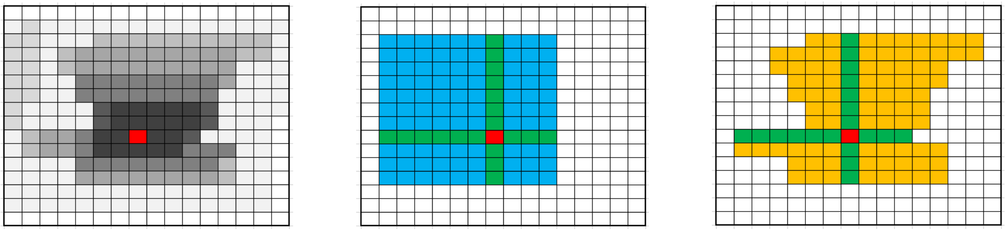
The aforementioned advances have led to local methods that have also achieved a competitive performance against the global methods, while being faster alternatives. Yet, their computational demand makes most of them still far from applicable to many real-time applications. As mentioned previously, the filtering procedure given by Eq. 2.7 has complexity independent of the window size provided that integral images are used. However, integral images cannot be applied to weighted averaging procedure presented in Eq. 2.8. Consequently, the complexity of the local methods performing EAF is $\mathcal{O}(W \cdot H \cdot N \cdot D)$, where W and H are the width and height of the support kernel; and whereas, N and D are the number of pixels in the image and disparity ranges, respectively. The complexity can be reduced if small support windows are used. Yet, the window size plays a significant role in estimating disparities accurately. Generally, large windows are preferred in order to handle weakly textured regions and this results in a comparable processing time for stereo pair images to global methods. This problem has led to stereo matching literature branching out into reducing the complexity of the EAF based local methods.

2.3.1 Reduction in Computational Complexities

There have been studies to reduce the complexity of the local methods. These methods can be classified according to intended source of complexity. As mentioned in Sec. 1.2.1, there are two factors of complexity except the image size. Therefore, the methods can be classified into three which are the methods aiming complexity reduction based on window size [17,21,35,49,60,75,81,82,91], methods aiming complexity reduction based on disparity search range [9, 36, 37, 66] and methods aiming complexity reduction on both [50, 55, 56]. The methods focusing on complexity reduction based on the window size exploit the regular grid structure of cost volumes and perform EAF in an efficient manner. These algorithms achieve window size independent complexity by getting rid of repetitive calculations via systematic operations that reuse the shared computation among neighbouring pixels. The second class of methods which decrease disparity search range either follows coarse-to-fine (CTF) [36, 37, 66] or stochastic approaches [9]. The main challenge is to reduce the complexity due to both the disparity range and the kernel size jointly. Irregular data access scheme of CTF [36,37,66] or stochastic methods [9,50] is highly opposed to regular and systematic computing style of the filtering methods that can reduce the complexity due to the kernel size [17, 35, 60]. Therefore, the approaches to reduce the complexity due to the disparity range introduces kernel size dependent complexity that cannot be reduced by the efficient filtering methods available in the literature [17, 35, 60]. The third class of algorithms deal with this problem [50, 55, 56] via stochastic [50] or sampling based [55,56] approaches. In the following subsections, the advances to solve the complexity reduction problem are presented with some detail on leading algorithms and a representative comparison is provided in Table 2.1.

2.3.1.1 Complexity Reduction on Window Size

Prior to bilateral filtering, **color adaptive windowing** is utilized in [75] to adapt size and location of the window according to color intensities of the pixel cost of which is to be filtered. Being one of the fastest methods, this method is worth to examine. According to a predefined color similarity threshold, T, four furthest pixels of a rectangular window is determined in horizontal and vertical directions within a given length limit, L. The bounds of the filtering window is determined from these pixels as shown in Fig. 2.4b. Once the window is specified, the box filtering is performed using integral images in a constant time. Despite the quite efficient filtering, this technique does not perform color adaptive averaging which is the key feature of bilateral filtering. It performs uniform cost aggregation within the window instead.



(a) (b) Adaptive Box (c) Arbitrary Shaped Cross

Figure 2.4: Comparison of the support regions resultant from color adaptive windowing and arbitrary shaped windowing for the local region of the pixel marked by red in (a).

Arbitrary shaped cross filter [91], which is similar to color adaptive windowing, is proposed to perform filtering over arbitrarily shaped windows instead of rectangular shaped windows. Similar to color adaptive windowing, four furthest pixels are determined for each pixel. Then using integral images formed for each horizontal scanline, the cost aggregation is performed according to horizontal bounds for each pixel. This aggregation is followed by vertical aggregation in a similar way on the updated data. The shape of a support region is shown in Fig. 2.4c. The support window of this approach is consistent with the connectedness of support pixels. This approach is similar to the geodesic support filtering in this manner but shares the same drawback with color adaptive windowing due to non color adaptive support weights.

After Bilateral Filtering is introduced to the stereo matching problem [74], several studies have been performed to reduce its complexity dependency on the window size [24, 30, 58, 83]. As a powerful efficient alternative edge-aware filtering to Bilateral Filter, *Guided Filter* [30] is proposed whose complexity is independent of the kernel size and it is improved in [21, 49] for better stereo matching purposes. All of these techniques work on a predefined window and they are approximations of bilateral filtering. Recently, recursive filtering approaches [17, 27, 61, 82] have been proposed, noted as the fastest edge-aware filters with complexities independent of the kernel size. They can be viewed as approximations of geodesic support filtering. Unlike previous approaches, they do not require preprocessing to provide window size independent complexity and they do not require predefined window size for filtering.

Guided Filtering

In [30], EAF given by Eq. 2.9 is considered as linear translation-variant filtering. The matching cost volume to be filtered is denoted as input and the reference image function is denoted as *guidance image.* With these terms, the filter can be viewed as a locally linear filter. Following this observation, the key assumption leading to guided filtering is the existence a local linear model between guidance image, I_r, and filtered data C_f, as,

$$C_f[\bar{y}, d] = a_{\bar{x}} I_r[\bar{y}] + b_{\bar{x}}, \ \forall \bar{y} \in \mathcal{N}_{\bar{x}} \ , \tag{2.16}$$

where $\mathcal{N}_{\bar{x}}$ represents rectangular local region centred at $\bar{x}$ and $(a_{\bar{x}}, b_{\bar{x}})$ are some linear coefficients assumed to be constant in $\mathcal{N}_{\bar{x}}$. This locally linear model enforces the cost variation within a cost slice to be consistent with color intensity variation within the guidance image in a local region, $\mathcal{N}_{\bar{x}}$, so that the filtered data has an edge if the guided image also has an edge: $\bigtriangledown C_f[\bar{y}] = a_{\bar{x}} \bigtriangledown I_r[\bar{y}]$. The filter coefficients $(a_{\bar{x}}, b_{\bar{x}})$ is determined by minimizing the difference between input data and the filtered data, as,

$$\begin{aligned} E(a_{\bar{x}}, b_{\bar{x}}) &= \sum_{\bar{y} \in \mathcal{N}_{\bar{x}}} \left[(a_{\bar{x}} I_r[\bar{y}] + b_{\bar{x}} - C[\bar{y}, d])^2 + \epsilon a_{\bar{x}}^2 \right] \\ (a_{\bar{x}}, b_{\bar{x}}) &= \arg\min_{(a,b)} E(a, b) \ . \end{aligned} \tag{2.17}$$

In [30], the solution is obtained as,

$$\begin{aligned} a_{\bar{x}} &= \frac{\dfrac{1}{|\mathcal{N}_{\bar{x}}|} \displaystyle\sum_{\bar{y} \in \mathcal{N}_{\bar{x}}} I_r[\bar{y}] \cdot C[\bar{y}, d] \ - \mu_{\bar{x}} \underline{\mathrm{C}}_{\bar{x},d}}{\sigma_{\bar{x}}^2 + \epsilon} \\ b_{\bar{x}} &= \underline{\mathrm{C}}\bar{x}, d - a_{\bar{x}} \mu_{\bar{x}} \ , \end{aligned} \tag{2.18}$$

where $\mu_{\bar{x}}$ and $\sigma_{\bar{x}}^2$ are the mean and the variance of the guidance image in $\mathcal{N}_{\bar{x}}$, $|\mathcal{N}_{\bar{x}}|$ is the number of pixels in the filtering window, $\underline{\mathrm{C}}_{\bar{x},d}$ is the mean of the filter input in $\mathcal{N}_{\bar{x}}$ and ϵ is a regularizing parameter to prevent $a_{\bar{x}}$ from becoming too large. The coefficients in Eq. 2.18 are obtained for just one local region around a pixel. However, the relation in Eq. 2.16 should hold for every region containing pixel $\bar{y}$. This implies that the pixel $\bar{y}$ gets a different value for every region containing it. The question here is which value should be assigned. In [30], this problem is handled by averaging the values of the coefficients coming from the local regions containing the particular pixel. In this way, the constraints on coefficients coming from different regions are merged. Thus, the guided filtering finally becomes:

$$C_f[\bar{y},d] = \sum_{\bar{x}:\bar{y}\in\mathcal{N}_{\bar{x}}} \left(a_{\bar{x}} I_r[\bar{y}] + b_{\bar{x}}\right) \ . \tag{2.19}$$

The guided filter defined through Eqs. 2.18 and 2.19 can be efficiently implemented by performing box filtering, namely, exploiting integral images. Hence, the complexity becomes independent of the filter kernel size. Only a specific number of operations are required for each pixel and this makes the filter complexity linear with image size. By the convention given in Eq. 2.8, adaptive weights of the guided filter can be expressed as,

$$w_{\bar{x},\bar{x}'} = \frac{1}{|\mathcal{N}|} \sum_{\bar{y}:(\bar{x},\bar{x}')\in\mathcal{N}_{\bar{y}}} \left(1 + \frac{\left(I_r[\bar{x}] - \mu_{\bar{y}}\right)\left(I_r[\bar{x}'] - \mu_{\bar{y}}\right)}{\sigma_{\bar{y}}^2 + \epsilon}\right) \ , \tag{2.20}$$

in order to observe edge preserving characteristic of it. In Fig. 2.3, the effective weights of the guided filter is provided for a particular pixel. It can be observed that color-wise similar pixels in a local region give more support. Guided filtering is a powerful and a sate-of-art approximation to the bilateral filtering. Similar to bilateral filtering, guided filtering does not consider connectedness among supporting pixels as well.

Recursive Edge Aware Filtering Recursive edge aware filtering techniques have been proposed in independent studies [17, 27, 61, 82] and reported as the fastest EAF methods. Such independently proposed filters are quite similar to each other and their basic scheme to perform EAF is to apply the following 1-D recursive and progressive filters in horizontal and verticals directions of the 2-D input data:

$$\begin{aligned} y_r[n] &= x[n] + \mu\, y_r[n-1] \\ y_p[n] &= x[n] + \mu\, y_p[n+1] \\ y[n] &= \frac{y_r[n] + y_p[n] - x[n]}{\eta} \ . \end{aligned} \tag{2.21}$$

Here $x[n]$ is the input data, $y[n]$ is the filtered output and η is some normalization constant which is explained later. If the coefficient μ, $0 \leqslant \mu < 1$, is kept constant, the filtering becomes linear and time-invariant. Investigation of Fourier transform of the transfer function of the filter in Eq. 2.21 gives insight about its behaviour. If the transfer functions of the recursive and progressive filters given by Eq. 2.21 are denoted as $h_r[n]$ and $h_p[n]$, respectively, the filters can be expressed as,

$$\begin{aligned} y_r[n] &= x[n] + \mu\, y_r[n-1] = h_r[n] * x[n] \\ y_p[n] &= x[n] + \mu\, y_p[n+1] = h_p[n] * x[n] \ , \end{aligned} \tag{2.22}$$

and Fourier transform of these transfer functions are

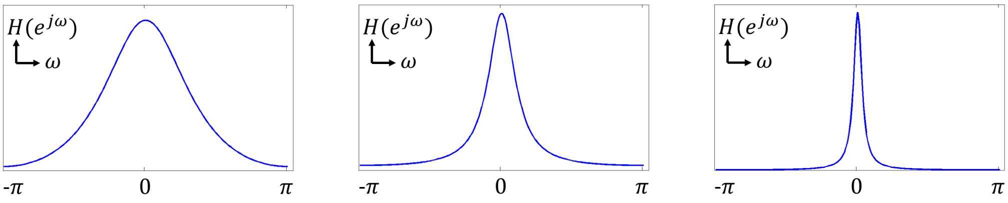
$$\begin{aligned} H_r(e^{j\omega}) &= \frac{1}{1-\mu \cdot e^{-j\omega}} \\ H_p(e^{j\omega}) &= \frac{1}{1-\mu \cdot e^{j\omega}} \ . \end{aligned} \tag{2.23}$$

Using linearity property of Fourier transform, Fourier transform of $y[n]$ in Eq. 2.21 can be written as,

$$Y(e^{j\omega}) = \frac{(H_r(e^{j\omega}) + H_p(e^{j\omega}) - 1)}{\eta} X(e^{j\omega}) \ . \tag{2.24}$$

From Eqs. 2.23 and 2.24, Fourier transform of the transfer function of the filter in Eq. 2.21 becomes

$$\Rightarrow H(e^{j\omega}) = \frac{\lambda}{1 + \dfrac{4 \cdot \mu}{(1-\mu)^2} \ \sin^2(\dfrac{\omega}{2})}, \ \lambda = \frac{1+\mu}{(1-\mu)\ \eta} \ . \tag{2.25}$$



(a) $\mu = 0.3$ (b) $\mu = 0.7$ (c) $\mu = 0.9$

Figure 2.5: Low-pass filtering characteristic of the filtering presented in Eq. 2.21. As μ increases, so does smoothing.

Examining the transfer function derived in Eq. 2.25 and the plots provided in Fig. 2.5, the low pass nature can be observed. The amount of smoothing depends on the value of the filter coefficient μ. This situation is shown in Fig. 2.5. As μ increases, the smoothing also increases and as μ goes to zero, no smoothing is imposed. In recursive edge aware filtering, it is proposed to control this behaviour by adapting the filter coefficients to the similarity between two adjacent input signal samples in a particular direction. That is to say,

$$\begin{aligned} y_r[n] &= x[n] + \mu(x[n], x[n-1])\, y_r[n-1] \\ y_p[n] &= x[n] + \mu(x[n], x[n+1])\, y_p[n+1] \\ y[n] &= \frac{y_r[n] + y_p[n] - x[n]}{\eta} \ . \end{aligned} \tag{2.26}$$

Here $\mu(x[n], x[m])$ becomes the adaptive filter coefficient, which is a monotonic decreasing function of the dissimilarity between $x[n]$ and $x[m]$. For the rest of the thesis, $\mu(x[n], x[m])$ is denoted as $\mu_{n,m}$ for the sake of notational simplicity. In [17, 27], this function is given as,

$$\mu_{n,m} = \mathrm{e}^{-\dfrac{|x[n] - x[m]|}{\sigma}} \ , \tag{2.27}$$

where σ is the decay control parameter similar to the ones in Eq. 2.11 that controls the smoothing based on the similarity. The term $\mu_{n,m}$ is referred as the *permeability coefficient* in [17] given that n and m are spatially adjacent. The term comes from the weak analogy of the term *permeability* used in biomedical engineering which defines the percentage of the molecules that can pass through the cell membranes. In a similar manner, permeability coefficients indicate the degree of the influence between adjacent locations during the recursive filtering process.

Edge preserving behaviour of the recursive filter given by Eq. 2.26 can be proved by expressing the filter with the same convention as in Eq. 2.8. Given an 1-D input signal of length L and starting from $n = 1$, by following the derivation procedure presented in [17], the recursive and progressive filters can be expressed as,

$$\begin{aligned} y_r[n] &= x[n] + \mu_{n,n-1}\, y_r[n-1] \\ &= x[n] + \mu_{n,n-1}\left\{x[n-1] + \mu_{n-1,n-2}\, y[n-2]\right\} \\ &\vdots \\ y_r[n] &= x[n] + \sum_{k=1}^{n-1}\Big(x[n-k]\underbrace{\prod_{l=1}^{k}\mu_{n+1-l,n-l}}_{w_{n,n-k}}\Big) \end{aligned} \quad . \tag{2.28}$$

A similar derivation can be made for progressive filtering, yielding,

$$y_p[n] = x[n] + \sum_{k=1}^{L-n}\Big(x[n+k]\underbrace{\prod_{l=1}^{k}\mu_{n-1+l,n+l}}_{w_{n,n+k}}\Big)\ . \tag{2.29}$$

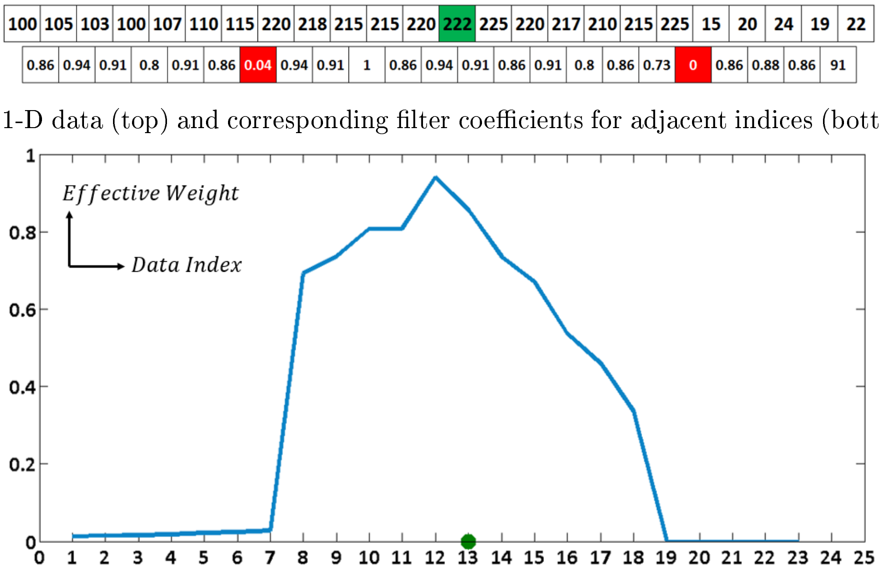
Using the results of Eqs. 2.28 and 2.29, the 1-D filter presented in Eq. 2.26 can be expressed as,

$$y[n] = \frac{\sum_{m=1}^{L}\left(w_{n,m}\, x[m]\right)}{\eta}, \tag{2.30}$$

where

$$\begin{aligned} \eta &= \sum_{m=1}^{L} w_{n,m} \\ w_{n,m} &= \begin{cases} \prod_{k=m}^{n}\mu_{k+1,k}, & m < n \\ 1, & m = n \\ \prod_{k=n}^{m-1}\mu_{k,k+1}, & m > n \end{cases} \quad . \end{aligned} \tag{2.31}$$

From Eq. 2.31, it can be concluded that when determining the filter output at an instant $y[n]$ according to weighted averaging, the weight of a distant data $x[m]$ is the successive product of the permeability coefficients from point m up to n in the corresponding direction. This implies that if a discontinuity is present in the input data between m and n then the influence of $x[m]$ cannot be transferred to the point m. Consequently, $x[m]$ cannot make a contribution to the averaging. It can be deduced that if a discontinuity exists in the preceding data according to data of interest, then the influence beyond that discontinuity cannot be passed to the data of interest. The same is also true for subsequent data. An illustration of these explanations is provided in Fig. 2.6. In Fig. 2.6 a 1-D input signal and the corresponding filter coefficients for adjacent indices are given. According to these, the effective weight distribution of an index is provided. From this distribution, the edge aware characteristic of the filter given by Eq. 2.26 can be observed.



100	105	103	100	107	110	115	220	218	215	215	220	222	225	220	217	210	215	225	15	20	24	19	22
0.86	0.94	0.91	0.8	0.91	0.86	0.04	0.94	0.91	1	0.86	0.94	0.91	0.86	0.91	0.8	0.86	0.73	0	0.86	0.88	0.86	91	

(a) 1-D data (top) and corresponding filter coefficients for adjacent indices (bottom)

(b) Distribution of the effective weights

Figure 2.6: Effective weight distribution for the data index 13 (marked by green in (a)) after performing 1-D filtering presented in Eq. 2.26.

A closer look at the effective coefficients in Eq. 2.31 in terms of weighting function in Eq. 2.27 also gives insight into the edge aware behaviour of the filter:

$$w_{n,m} = \begin{cases} \mathrm{e}^{-\frac{\sum\limits_{k=m}^{n} |x[k+1]-x[k]|}{\sigma}}, & m < n \\ 1, & m = n \\ \mathrm{e}^{-\frac{\sum\limits_{k=m}^{n} |x[k]-x[k+1]|}{\sigma}}, & m > n \end{cases} . \tag{2.32}$$

The exponent here is the sum of absolute differences of the adjacent data between the two indices. This sum can be considered as the geodesic distance between the two indices once the distance between the two indices is defined according to the data differences at the adjacent indices between them. In other words, the distance between the two indices is defined as the length of the trace of the data between them, as shown in Fig. 2.7. Despite the fact that actual length consists of spatial and data difference components as depicted in Fig. 2.7, it can be assumed that spatial distance is negligible and the length can be defined based on the data differences only. This definition of the distance is similar to the path cost definition in Eq. 2.12. From these conclusions, the analogy between the recursive filter defined in Eq. 2.26 and the geodesic support filter can be drawn, and this proves the edge aware behaviour of the proposed filters in [17, 27]. Actually, the work in [27] has emerged from this point. The authors of [27] have sought to find an efficient way to determine the geodesic distance between the two points for EAF and ended up with the filter presented in Eq. 2.26.

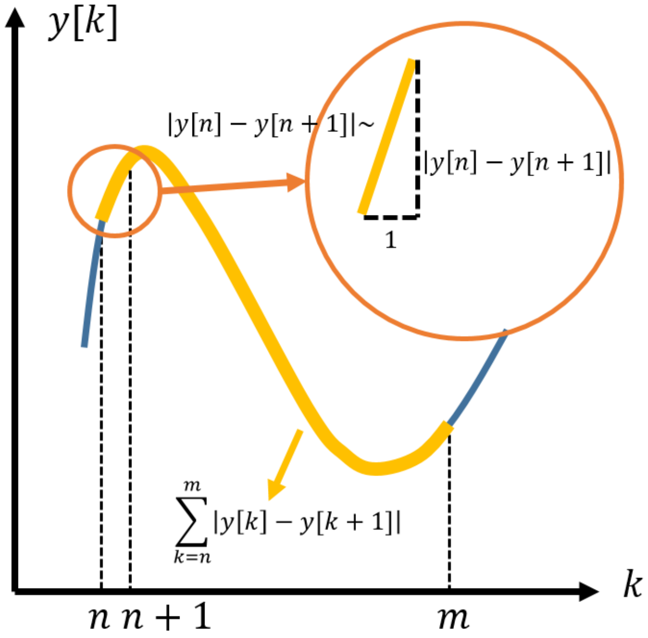


Figure 2.7: Curve length between two indices (n, m). If the spatial displacement between two adjacent indices in the data is neglected, then the curve length becomes sum of absolute data differences.

The discussion of the edge aware recursive filtering is based on 1-D data so far. In order to perform 2-D filtering, *2-pass 1-D filtering*, which is sequential application of the 1-D filter in Eq. 2.26 in horizontal and vertical directions with some modifications, is proposed in [16, 17, 27, 60, 61, 82]. It is important to note that the modification is related to normalization coefficient, η, which is explained soon.

In [17], two horizontal passes are followed by two vertical passes on the updated data by applying the filter in Eq. 2.26 without any normalization. The normalization is performed after the passes are completed by dividing each filtered value by the sum of the effective weights of the data influenced that value. That is, horizontal passes are performed on 2-D input data, $I[\bar{x}]$, first, as,

$$\begin{aligned} I_r[\bar{x}] &= I[\bar{x}] + \mu_{\bar{x},\bar{x}-\bar{1}_h}\, I_r[\bar{x} - \bar{1}_h] \\ I_p[\bar{x}] &= I[\bar{x}] + \mu_{\bar{x},\bar{x}+\bar{1}_h}\, I_p[\bar{x} + \bar{1}_h] \\ I^h[\bar{x}] &= I_r[\bar{x}] + I_p[\bar{x}] \ , \end{aligned} \tag{2.33}$$

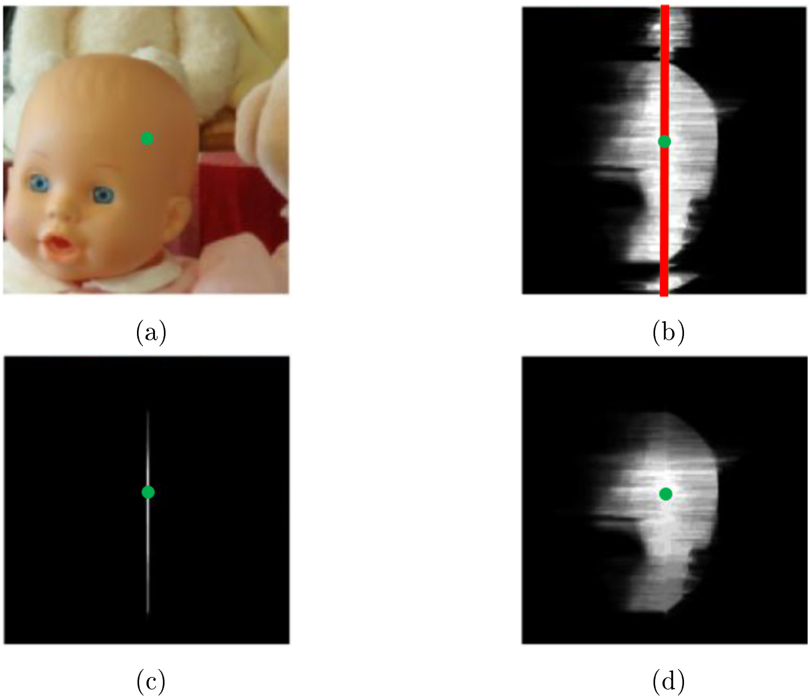
where $\bar{x} = (x, y)$ and $\bar{1}_h = (1, 0)$. After the horizontal passes, vertical passes are performed on updated data, I^h, as,

$$\begin{aligned} I^h_r[\bar{x}] &= I^h[\bar{x}] + \mu_{\bar{x},\bar{x}-\bar{1}_v}\, I^h_r[\bar{x} - \bar{1}_v] \\ I^h_p[\bar{x}] &= I^h[\bar{x}] + \mu_{\bar{x},\bar{x}+\bar{1}_v}\, I^h_p[\bar{x} + \bar{1}_v] \\ I^v[\bar{x}] &= I^h_r[\bar{x}] + I^h_p[\bar{x}] \ . \end{aligned} \tag{2.34}$$

Similarly, $\bar{1}_v = (0, 1)$. Finally, the filtered data is obtained after the normalization. Each value of I^v is divided by the sum of the effective weights of the data influenced I^v. This is efficiently performed by the filtering process on a dummy data consisting of ones so that accumulated values correspond to sum of effective weights of the distant points, thanks to ones being identity element of multiplication operation. If the filtering process given in Eqs. 2.33 and 2.34 is denoted as $\mathcal{F}$ such that $\mathcal{F}(I) = I^v$, then the normalized filtering result, I_f, can be obtained as,

$$I_f = \frac{\mathcal{F}(I)}{\mathcal{F}(1)} \ . \tag{2.35}$$

After the two passes given in 2.33 and 2.34 sequentially, the effective weight of a point $\bar{y}$ with respect to a reference point $\bar{x}$ becomes the successive product of the permeability coefficients until reaching the reference point by proceeding on 1 horizontal and 1 vertical direction. In this way, support from a distant point can



(a)

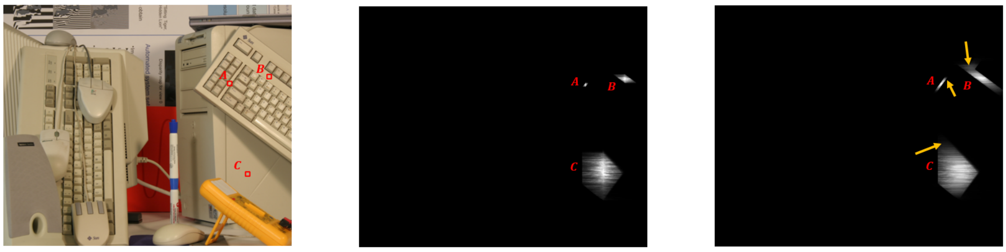
(b)

(c)

(d)

Figure 2.8: Effective support weights for the pixel indicated by greed dot in (a) after performing 2-pass 1-D filtering proposed in [17]. (b) Effective weights of the pixels with respect to the vertical line passing through the pixel of interest after left-to-right and right-to-left passes. (c) Effective weights of the pixels with respect to the pixel of interest after top-to-bottom and bottom-to-top passes. (d) Effective support weights after performing the horizontal and vertical passes sequentially on the updated data. Support regions are retrieved from [16].

be obtained in the 2-D domain. The effective weights according to a reference point is illustrated in Fig. 2.8. The reference point is shown by the green dot. After the horizontal passes, the effective weights of the distant points according to the points on the vertical line where the reference point lies are shown in Fig. 2.8b. The effective weights of the points on the vertical line are shown in Fig. 2.8c according to vertical passes. The effective weights of the distant points after horizontal and vertical passes are shown in Fig. 2.8d. As it can be observed from the figure, 1-D filtering in horizontal and vertical directions sequentially provides arbitrarily shaped weighted support regions. This result can be viewed as an approximation to the geodesic support region. Geodesic paths of the support points are approximated by 1 horizontal and 1 vertical paths. Hence, only the support points that have the geodesic paths that can be well approximated by 1 horizontal and 1 vertical paths can influence the filtered value. For better approximations of the geodesic supports, multiple iterations consisting of horizontal and vertical passes can be performed as shown in Fig. 2.9.



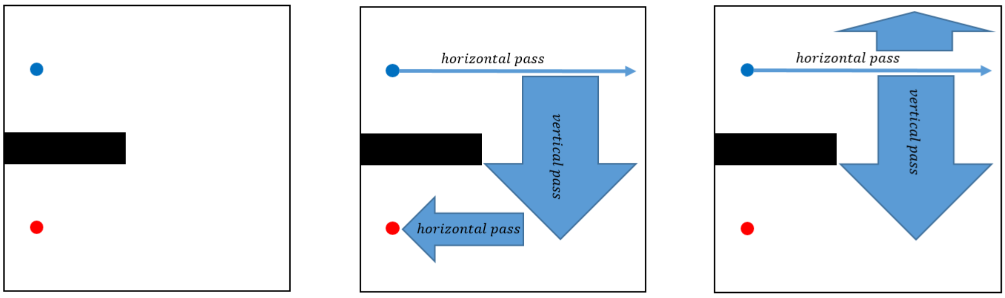
(a) (b) 1 iteration (c) 2 iterations

Figure 2.9: Effect of performing 2-pass 1-D filtering iteratively. The geodesic support weights are better approximated by multiple passes.

In [27], another approach, which is slightly different from the approach explained above [17], is proposed for 2-pass 1-D filtering. The difference is the application of 1-horizontal pass and then 1-vertical pass to the updated data. In this way, by performing 4 passes, the geodesic support region can be approximated better compared to 2-pass horizontal and 2-pass vertical approach [17]. Fig. 2.10 illustrates the differences between two approaches in terms of influence diffusion technique. Apart from these vertical and horizontal pass based approaches, a tree based approach is proposed in [81] in order to improve the influence diffusion further. The 1-D filtering is performed on the minimum spanning tree extracted from the reference image instead of along the horizontal and vertical directions. In [54], this tree based approach is also utilized in an over-segmented image representation based method. Similar ideas are previously used in semi-global methods [54, 81].

The above-mentioned filtering techniques are applied to cost aggregation step of local stereo matching methods via joint filtering. In joint filtering, the input data is a 2-D slice of the matching cost volume, C, under a fixed disparity, d, and the adaptive filter coefficients are determined according to reference image function, I. General weighting function used in studies [17, 60] is

$$\mu_{\bar{x},\bar{y}} = \mathrm{e}^{-\dfrac{max\left\{|I^R[\bar{x}] - I^R[\bar{y}]|, |I^G[\bar{x}] - I^G[\bar{y}]|, |I^B[\bar{x}] - I^B[\bar{y}]|\right\}}{\sigma}}, \qquad (2.36)$$



(a) (b) Domain Transform (c) Inf. Perm.

Figure 2.10: Comparison of the different 2-pass 1-D filtering techniques utilized by Domain Transform based [27] and Information Permeability [17] based filtering. (c) treats the input data symmetrically, whereas (b) follows an asymmetric strategy enabling approximation of more complex paths. (b) can only diffuse the influence as long as there is an L path between pixels.

where I^R, I^G, I^B represent the red, green and blue color channels of the reference image, I, and the cost aggregation is,

$$C_f[\bar{x}, d] = C[\bar{x}, d] + \mu_{\bar{x},\bar{x}-\bar{1}_{pd}} C_f[\bar{x} - \bar{1}_{pd}, d] \ . \tag{2.37}$$

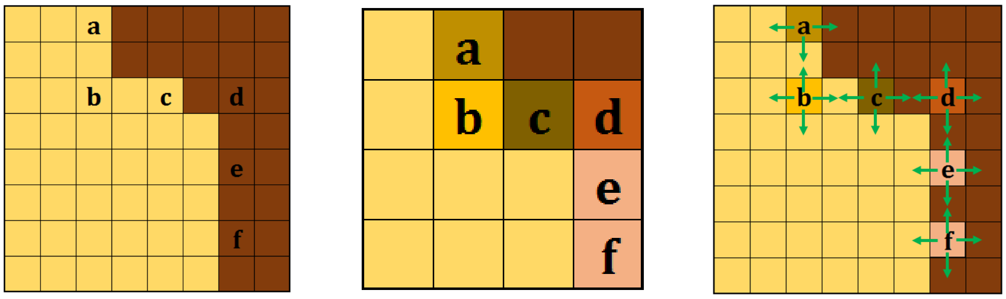
Here $\bar{x} - \bar{1}_{pd}$ represents the previous pixel in the particular passing direction.

The cost aggregation strategy as given by Eq. 2.37 is reported as the fastest discontinuity preserving cost filtering technique due to its quite limited number of multiplication and addition operations per pixel [17]. As mentioned previously, these recursive filtering approaches approximate geodesic support kernels so that connectedness among supporting pixels and the reference pixel is preserved unlike the bilateral filtering approaches. Moreover, recursive structure enables expansion of the filter kernel to the entire image; thus, weak textured or untextured regions can be handled better without increasing the complexity due to windows size. In fact, no predefined window size is required; this is one of the most distinctive features of edge aware recursive filtering. On the other hand, recursive structure introduces a great weakness in the presence of noise or high texture. Only one noisy pixel on the L path composed of horizontal and vertical paths prevents the influence from diffusing between two pixels due to successive and regular transfer of influence.

2.3.1.2 Complexity Reduction on Disparity Search Range

The common property of the methods that reduce complexity in disparity search range is the search space reduction via determining some candidate disparities for each pixel. According to candidate disparity generation techniques, there are three main classes of approaches aiming to reduce complexity present in disparity search range. These techniques are coarse-to-fine (CTF) methods [36, 37, 66], sampling based approaches [55, 56] and stochastic methods [9, 50].

Coarse-to-fine methods [36,37,66] estimate disparities at a coarse resolution and utilizes these estimates to reduce their search range at the finer resolutions. Image pyramids and hierarchical estimation of disparity from coarser scales to finer scales theoretically decreases the complexity in the search range. However, this reduction in complexity might yield some performance drop. Due to the low pass filtering during image down-sampling, some details presented in the image might be lost and the estimates near object boundaries become weak. Considering Fig. 2.11, the pixels, (a, b, c, d, e, f), are near an object boundary at the fine resolution (Fig. 2.11a). When the coarse scale image is built, the foreground and background data are merged due to low pass filtering during downsampling (Fig. 2.11b). Consequently, during disparity estimation, these pixels are generally mismatched. They are either assigned to the wrong background or foreground disparity or assigned to another disparity somewhere between background and foreground. When the estimates of these pixels are used to reduce the disparity search range at the finer scale (Fig. 2.11c), they mislead the neighbouring pixels during the correspondence search. The propagation of those erroneous estimates can never be recovered in the finer scales. These problems are tried to be addressed in [36, 37, 66]. In [37], the fine structure loss problem is addressed and different coarse scales are used for different regions according to their structure. A saliency analysis is performed to determine the amount of downsampling for each region. In [36, 66], transfer of disparity information from a coarse scale to the finer scale is addressed. In [36,37], the disparity candidate set for each pixel, $\mathcal{C}_{\bar{x}}$, is formed by picking all the disparities that are within some distance, γ, to the estimated disparities of the pixels lying on a local neighbourhood. Explicitly,



(a) Fine scale (b) Coarse scale (c) Coarse-to-fine scale

Figure 2.11: Illustration of the reasons of edge-fattening problem in CTF approaches. For downsampled image in (b), the pixels (a, b, c, d, e, f) get information from both foreground and background. Therefore, their estimates are generally poor. Transferring erroneous estimates to fine scales causes propagation of the error and results in poor performance at the object boundaries as shown in (c).

if $\mathcal{N}_{\bar{x}}$ is denoted as local neighbourhood of pixel $\bar{x}$ and $\mathcal{D}_{\mathcal{N}_{\bar{x}}}$ is denoted as the set of disparities belonging to the pixels in the neighbourhood that are estimated at coarse scale, then the candidate set becomes:

$$\mathcal{C}_{\bar{x}} = \{d \mid min\,(|d - 2\,\mathcal{D}_{\mathcal{N}_{\bar{x}}}| \leqslant \gamma)\} \quad . \tag{2.38}$$

In Eq. 2.38, the multiplication by 2 comes from scaling the estimation value at the coarser scale. The realization of the process is depicted in Fig. 2.12. The 4

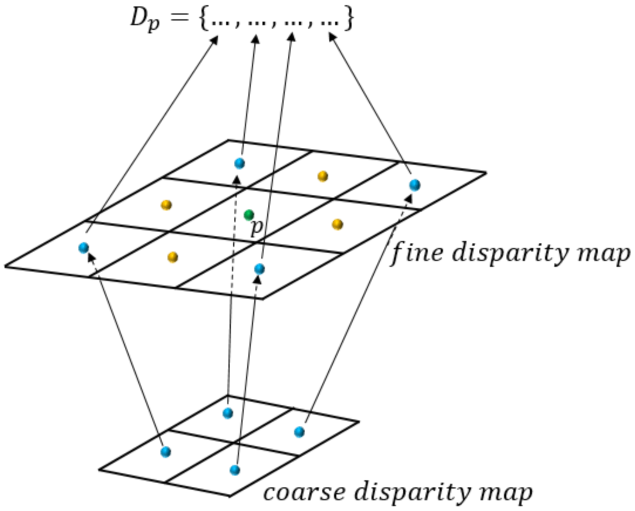


Figure 2.12: Disparity candidate set formation. The estimates at coarse scale are utilized to form the candidate set for pixel p. Estimates of nearest 4-neighbours of p are used.

nearest neighbours of the pixel p are marked by the blue dots at the fine scale image. These neighbours have disparity estimates that are available from the coarse scale. The candidate set of p is formed according to Eq. 2.38 by using the disparity estimates of these nearest 4 neighbours. Once the disparity candidate set is formed, cost aggregation is performed for the matching cost of each candidate. This process introduces additional computations when disparities in the candidate set are distinct. In [66], this problem is dealt with by using an approach similar to shifted windows [38]. In that method, the candidate set is formed in two steps. In the first step, each pixel at finer disparity forms its candidate set by using the disparity estimation of its nearest neighbour from the coarser scale. In the second step, following to disparity assignment via simple box filtering, final candidate set is formed by taking the disparity estimation of the neighbour with minimum cost.

Different from CTF approaches, a sampling based method is proposed in [55,56]. The redundancy in the disparity search space is reduced through a preprocessing step. The matching costs is filtered by box filters and the candidates are determined as the local optimum points. Finally, stochastic approaches [9, 50] have been developed. They exploit *Patch Match* method [3] in correspondence search and reduce the number of disparities in the disparity search space exponentially.

2.3.1.3 Complexity Reduction on Both Kernel Size and Disparity Search Range

As mentioned in the previous sections, simultaneous reduction of the complexity presented in both filter kernel size and disparity search range is a challenging problem. This challenge is recently attacked in [50, 55, 56]. In [50], a filter utilizing *Guided Filtering* [30] and *Patch Match* [3] method is presented. In order to filter irregular cost volume, over-segment representation of the image is used and the filtering for each segment is performed in each sub-images that covers the segment. Unlike usual local methods, this method has an iterative behaviour which cycles among the disparity hypotheses generation with the spirit of Patch Match method, disparity estimation and disparity estimate propagation to neighbouring segments. This method reduces complexity by introducing a preprocessing step and some modifications that actually prevent parallel implementation and degrade efficiency. In [55,56], a method based on reduction of the redundancies lying on disparity search range and filtering kernel is proposed. In those methods, the matching cost volume is pre-filtered with box filters, and then, the reduction on the disparity search space is achieved through considering only the disparities corresponding to local minimum points in the cost function for each pixel. Once the disparity candidates are specified, disparity assignment is performed via EAF based cost aggregation and WTA optimization. The redundancy in the filtering step is decreased by spatial sampling in the filtering kernel, resulting using only limited number of pixels for the cost aggregation instead of using all pixels in the windowed region.

Table2.1: Comparison of the state-of-the-art methods attacking the complexity reduction problem. The image size is denoted as N and the complexities due to window size and disparity search range are represented as W and D, respectively. Preprocessing represents whether a method requires an initial step to provide complexity reduction, such as over-segmentation. Rankings are according to execution times, smaller is better.

Method	**Complexity Reduction**		**Preprocessing**	**Time Complexity**	**Rank**
	W	**D**			
Guided Filtering	✓	×	✓	$\mathcal{O}(ND)$	3
Recursive EAF	✓	×	×	$\mathcal{O}(ND)$	1
CTF Approaches	×	✓	×	$\mathcal{O}(NW)$	4
Patch Match Filtering	✓	✓	✓	$\mathcal{O}(N\log(D))$	2

2.4 Cooperative Methods

Cooperative methods [8,19,59,71,79,92] exploit local and global optimization strategies together. The aim is to increase the estimation performance at the object boundaries and weakly textured regions and to handle occlusions better by unifying advantages of different techniques. The key assumption of cooperative methods is that a scene consist of non-overlapping surface patches and each patch corresponds to a pixel group of similar color. Smoothness is imposed on each segment and sharp disparity changes are allowed among segment boundaries. In this manner, cooperative methods are similar to region based global methods. Cooperative methods follow an iterative process. Disparities are estimated for each individual patch and then disparity information within each patch is propagated among neighbouring patches by constraining smoothness between color-wise similar neighbouring patches and penalizing occlusions and overlapping regions. This iterative behaviour of cooperative methods makes their computational complexity relatively high compared to other algorithms. On the other hand, this increase brings more precise disparity maps.

2.5 Occlusion Handling

Occlusion handling is the problem of estimating disparities in occluded regions and this problem is addressed in many stereo matching works. A review of these studies is also available in [13]. Disparity estimation problem for occluded regions is dealt with mainly in three ways. First one is implicit handling by leaving the problem to be solved by smoothness constraint [12,39,41,71]. This is the case when a global optimization framework is utilized. In [12,39], smoothness penalty term implicitly handles occlusions, whereas in [41, 71], region based approach provides disparity estimates at occluded regions.

The other two strategies are explicit handling during matching step or leaving the problem as a post processing step. The former is encountered in global [10,42,45,68,79,80,84] or semi-global [4,6,11,28,77] optimization based methods and the latter is a general approach for any class of methods; yet, it is mostly observed in local methods. In the methods that handle occlusion during matching step, an occlusion cost term is introduced to the cost function in Eq. 2.1. This is achieved by either embedding the occlusion cost in data term [10, 45, 80, 84] or explicitly defining a cost term related to occlusion [4, 6, 11, 28, 42, 68, 77, 79]. Then, the new cost function with the occlusion term is minimized as simultaneously detecting occlusions. In semi-global methods [4, 6, 11, 28, 77], occlusion label is added to the set of possible disparity assignments, so that occluded pixels can be labelled as "occluded" instead of a disparity value. Hence, as finding the minimum cost path, occlusions are detected and labelled by using ordering/monotonicity constraint. In [4,6], an occlusion constraint related with sharp color intensity changes is introduced in addition to the monotonicity constraint. In these approaches, occlusion handling is achieved by detection only; however, constraints imposed, such as finding the minimum cost path, enables disparity assignments to occluded regions implicitly. In global methods [10,42,45,79,80,84], either a cost term related to occlusions are added or the data term is modified so that a cost can be assigned to the pixels regarded as occluded. In both approaches, disparity assignment of occluded pixels become independent of matching costs. In this way, smoothness constraint is made more dominant for occluded pixels, enabling disparity estimation on occluded regions concurrently. Different from these approaches, the method proposed in [68] treats occlusion problem as an optimization problem. Occlusion map estimation is also considered together with disparity map estimation. Therefore, a global cost function of both occlusion map and disparity map is constructed and solved iteratively by fixing one map to obtain the other map.

Occlusion handling via post processing consists of two main steps which are occlusion detection and filling. The focus in this section is given only to the detection step. An empirical comparison of common occlusion detection approaches can be found in [23]. The occlusion detection can be performed either by using disparity map of a single image or by using disparity maps of the left and right images together. The former is referred as *asymmetric* [2,57,66,67,84,92] and the latter is referred as *symmetric* [6,11,15,28,51] occlusion detection.

Asymmetric methods make use of disparity map, matching costs, uniqueness and monotonicity constraints to detect occlusions. **Bimodality** which is proposed in [67] detect occlusions by examining the histogram of the disparities in local regions around pixels. A pixel is classified as occluded if its local histogram is a bimodal distribution. The assumption behind this reasoning is the local regions around occluded pixels have disparities coming from both background and foreground. **Goodness of match score** is an occlusion detection method based on matching costs. In [2], it is shown that matching costs of the algorithms can be used to detect occluded regions. According to goodness of match, pixels having weak match score are considered as occluded. The weak match score differs among algorithms; in [57, 66, 92], a weak match score corresponds to high matching costs in the minimum point of a 1-D cost function for a pixel, while in [84] it is considered as the weak distinctiveness of the minimum point according to other points in the 1-D function, as,

$$O[x,y] = \begin{cases} 1, & \tilde{C}[x,y,D[x,y]] > t \\ 0, & ow \end{cases} \tag{2.39}$$

$$O[x,y] = \begin{cases} 1, & \left| \dfrac{\tilde{C}[x,y,D_1[x,y]] - \tilde{C}[x,y,D_2[x,y]]}{\tilde{C}[x,y,D_2[x,y]]} \right| < t \\ 0, & otherwise \quad . \end{cases} \tag{2.40}$$

$\tilde{C}$ is the cost volume after some local or semi-global optimization process, $O[x,y]$ is an indicator function that takes 1 when a pixel at location (x,y) is occluded and t is some threshold to decide whether a pixel is occluded. In Eq. 2.40, D_1 and D_2 are the disparity maps obtained from the first and the second best costs respectively. Goodness based methods are applicable to stereo matching methods where disparity assignment is performed according to 1-D cost function of each pixel. **Constraint** based methods [4, 6, 11, 28, 57, 66] check the consistency between the disparity map and the imposed constraints. Uniqueness based methods warps reference disparity map to obtain other image's disparity map and then detects the pixels mapped to the same point. Among these pixels, the one with the largest disparity is picked as occluding and the others are detected as occluded pixels. If a warping function, $w[x,y,d] = (x^{'},y)$, is defined and the set of pixels that occlude the pixel at location (x,y) is denoted as $S[x,y]$, then occlusion detection method under uniqueness constraint becomes,

$$\begin{aligned} S[x,y] &= \Big\{(x^{'},y^{'}) \mid w[x^{'},y^{'},D[x^{'},y^{'}]] = w[x,y,D[x,y]] \\ &\quad \wedge D[x,y] < D[x^{'},y^{'}]\Big\} \\ O[x,y] &= \begin{cases} 1, & S[x,y] \neq \emptyset \\ 0, & otherwise \quad . \end{cases} \end{aligned} \tag{2.41}$$

Monotonicity based methods check the ordering constraint between the adjacent pixels and detect the pixels violating it. Detections of constraint based methods are valid, if the disparities of the pixels considered as visible are reliable. To make asymmetric occlusion detection more robust, methods using goodness and geometric constraints together are proposed [57, 66].

Symmetric methods require estimation of both disparity maps of the left and right image, and this doubles the complexity of the stereo matching algorithms. **Discontinuity** based methods [6, 11, 28, 51] use one disparity map to detect occlusions of the other disparity map via discontinuities in the depth map. It is assumed that left discontinuities in the right disparity map corresponds to the occluded regions in the left disparity map and vice-versa. **Left-right consistency checking** (LRC) [15] is a popular method which validates visibility of the pixels by cross-checking of the disparity maps. If the disparity of a pixel in one reference disparity map is different from the disparity of its correspondence in the other disparity map, then that pixel in the reference image is regarded as occluded. The situation for occlusion detection in left disparity map, D_l, using right disparity map, D_r, is:

$$O[x,y] = \begin{cases} 1, & \left|D_l[x,y] - D_r[x - D_l[x,y], y]\right| > t \\ 0, & otherwise \quad . \end{cases} \tag{2.42}$$

This method actually implies a uniqueness constraint, and therefore, similar to asymmetric method of uniqueness constraint. Therefore, in order not to increase the computational load, the asymmetric version is implemented. However, asymmetric handling makes erroneous decisions at slanted surfaces, while LRC can handle these situations by tolerating small disparity differences, t, between the two maps. This behaviour of LRC makes it good at eliminating bad disparity estimates and due to its simplicity together with this reason, it is implemented in many stereo matching algorithms.

CHAPTER 3

PROPOSED METHOD

In this section, a novel hierarchical stereo matching method with $\mathcal{O}(1)$[1] complexity is presented. Following the explanation of the motivation behind this work, the algorithm is presented in detail under two main sections. Next, the linear time complexity of the proposed method is theoretically proved and finally, the chapter is concluded by extensive tests on both disparity estimation accuracy under different circumstances and running time.

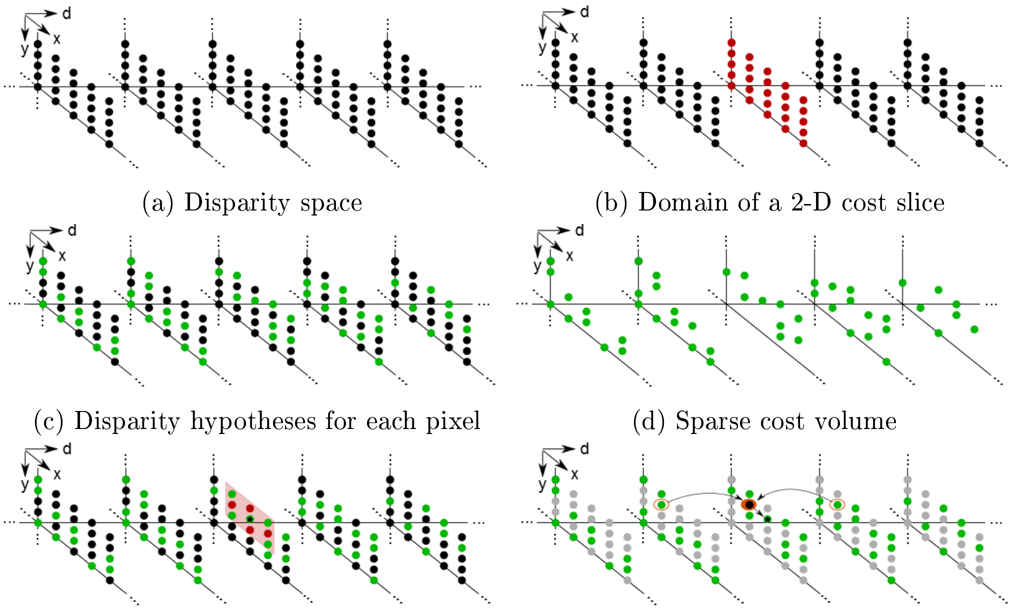
3.1 Motivation

The aforementioned approaches in Sec. 2.3.1.1 reduce the complexity of edge-preserving local stereo matching methods from $\mathcal{O}(N \cdot W \cdot D)$ to $\mathcal{O}(N \cdot D)$, which means the complexity due to cost aggregation step, W, is eliminated. Further reductions could only be related to disparity search range, D. Yet, it must be mentioned that the reduction of the complexities due to cost aggregation and disparity search range concurrently is a challenging task as a consequence of contradicting nature of the two objectives over the cost volume.

In general, as it is explained in Sec. 1.2.1, a local stereo matching algorithm performs a *matching cost calculation* (Eq. 1.3) and *cost aggregation* (Eq. 2.8) for every disparity value for every pixel. Cost aggregation can be viewed as filtering each 2-D slice of the cost volume under fixed disparity. Such a 2-D slice is illustrated in Fig. 3.1b. The methods focusing on reducing the computational

[1] What is meant by $\mathcal{O}(1)$ complexity here is linear complexity with image size.

load of cost aggregation step are based on using integral images [35, 75, 91] or recursive schemes [17, 60, 82]. Both of these approaches are based on calculation of aggregated cost of each disparity at each pixel and reusing the shared computation during cost aggregation.



(a) Disparity space

(b) Domain of a 2-D cost slice

(c) Disparity hypotheses for each pixel

(d) Sparse cost volume

(e) Support window and unknown costs

(f) Proposed cost prediction

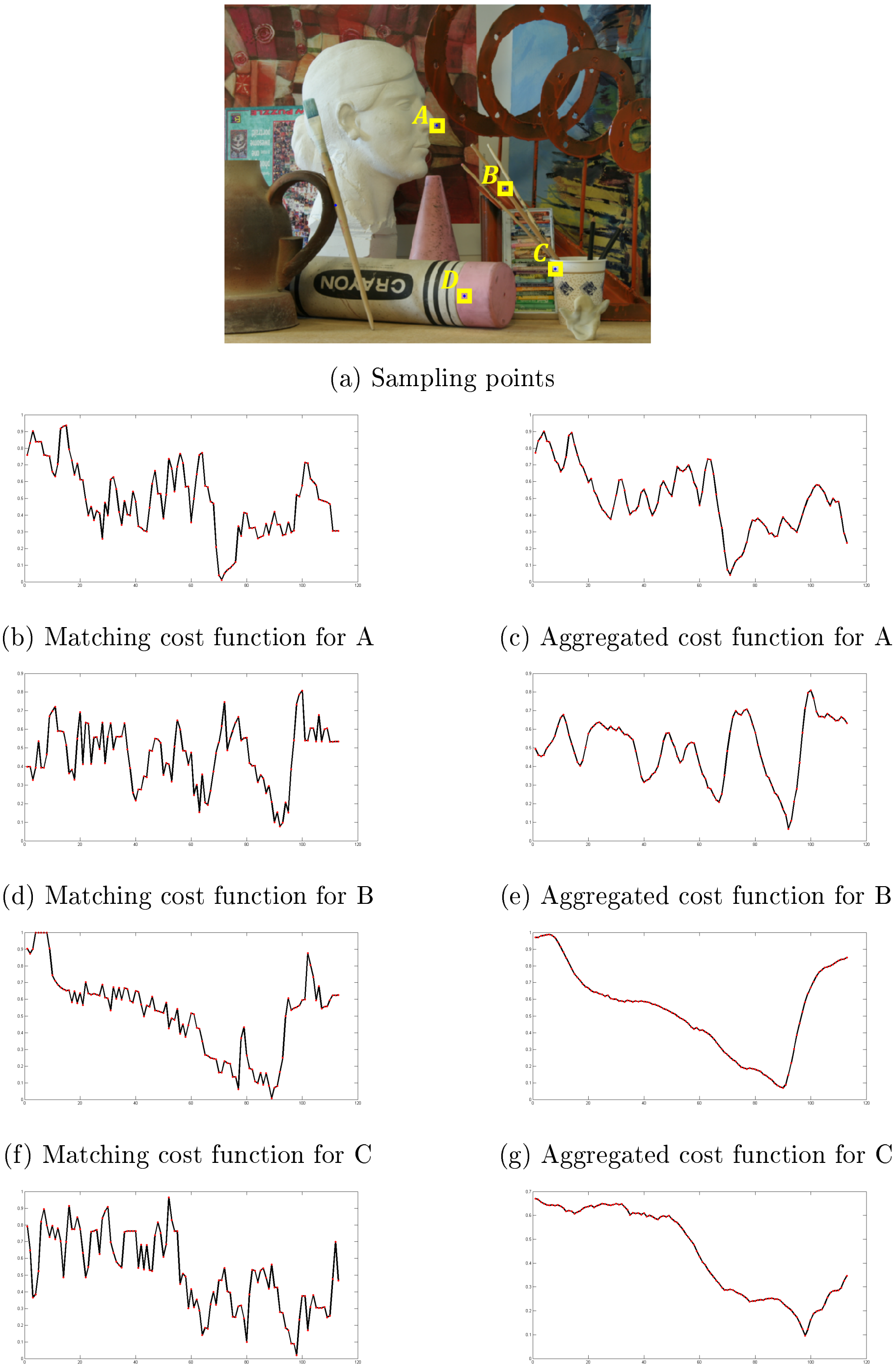
Figure 3.1: Aggregated cost volume formation for different methods.
In (b), cost aggregation is performed for each possible disparity at each pixel and another full cost volume is formed. The aggregation can be performed quiet efficiently in every 2-D slice of the cost volume (shown by red). In (c), every pixel has 2 disparity hypotheses and cost aggregation is performed only for those hypotheses, which results the sparse cost volume in (d) where aggregated costs of particular points are available. This sparse cost volume cannot be formed as efficiently as the cost volume in (b), since the aggregation methods applied in (b) cannot be applied. That methods make use of the aggregated cost values of the neighbouring points of the same disparity when dealing with a particular point, namely, the computations can be shared among neighbouring points as long as costs of the same disparity are considered. In (d), such a sharing is not possible, since no computations might be performed for neighbouring points during cost aggregation. Therefore, in order to form (d), cost aggregation is performed for each pixel individually within a window as in (e). The costs at the points marked by red are temporarily calculated for the sake of cost aggregation. In (f), the idea of utilizing the known aggregated cost values to estimate unknown cost value is depicted. In this way, the aggregated cost values of the neighbouring points can be useful for efficient cost aggregation.

On the other hand, the methods [36, 37, 50, 56, 66] aiming complexity reduction on disparity search range target the redundancy of the cost aggregation for every disparity value at each pixel. The main idea of these methods is to perform cost aggregation for only small number of disparity candidates which are determined somehow for each pixel, instead of for all disparities in the search range. In other words, the cost aggregation is to be performed for only the cost points marked by green in Fig. 3.1c; and this means only two disparity hypotheses are to be tested for each pixel in the image according to Fig. 3.1c. This situation results in a sparse cost volume[2] as shown in Fig. 3.1d, on which cost aggregation cannot be performed via 2-D filtering as in Fig. 3.1b. In other words, for a sole point in Fig. 3.1d, it is no more possible to calculate the aggregated cost value in a fast manner as in [17, 35].

It should be emphasized that the cost aggregation for the cost of a disparity assignment to a pixel is performed within a window. In this window, all costs values should be available prior to cost aggregation given Eq. 2.8. Therefore, the unavailable cost values should be computed temporarily by an extra effort for sparse cost volumes. This case is depicted in Fig. 3.1e where red marks represents those extra cost calculations for the cost aggregation of a particular pixel. In summary, reduction on disparity search range introduces window size dependent complexity whereas window size independent complexity requires cost aggregation for all disparity values due to inapplicability of the efficient filtering methods to sparse cost volumes.

This study is motivated by the question of whether it is possible to perform efficient filtering process given by Eq. 2.37 on sparse cost volumes by using the available cost values to predict the missing values. In other words, when the required aggregated cost of the preceding pixel at a disparity is unavailable, that cost is estimated from the known costs of the different disparities and it is used to update the aggregated cost of the current pixel. This situation is depicted in Fig. 3.1f. For a particular pixel, p, the required aggregated cost value of its preceding pixel, which is marked by black dot inside the orange

[2] The term "sparse" in this context is different from its general meaning which states having most of the elements zero value. In this context it is used to indicate that only limited cost values are available.



(a) Sampling points

(b) Matching cost function for A

(c) Aggregated cost function for A

(d) Matching cost function for B

(e) Aggregated cost function for B

(f) Matching cost function for C

(g) Aggregated cost function for C

(h) Matching cost function for D

(i) Aggregated cost function for D

Figure 3.2: Illustration of local linear behaviours of matching (Eq. 3.6) and aggregated (Eq. 2.37) cost functions of the pixels sampled from the image in (a) [62]

disk, is unavailable. This unavailable cost corresponds to disparity 3. If this unavailable cost is *well-predicted* from those of available 2 and 4, which are encircled by orange circles, in a *very simple manner*, then the predicted cost can be used to update the aggregated cost of the pixel, p, according to Eq. 2.37 without degrading the *efficiency.* Hence, the recursive filtering process given by Eq. 2.37 can be performed in the absence of the required cost values. In this way the reduction on disparity search range no longer introduces cost aggregation based complexity. It is argued that such an approach is possible under the following conditions and based on these assumptions, an efficient filtering method is proposed for sparse cost volumes:

1 Matching (Eq. 1.3) and aggregated (Eq. 2.8) cost functions of individual pixels can be well-approximated (predicted) efficiently within a small disparity range.

2 Colour-wise similar pixels in a local region have similar disparity hypotheses.

The first condition is required for *efficient* estimation of unknown cost values and it is backed by the assumption of linear varying of color intensities[3] employed in estimating the optical flow in several studies [69,73]. It should be noted that by the assumption of linear varying of color intensities in a local region of an image, estimation of cost values can be achieved in a linear form very efficiently; since, the matching costs given by Eq. 3.6 are based on color intensities. The efficiency in this context is considered as *fast* and *reliable* estimation of unknown cost values. In Fig. 3.2, local linear behaviour of several matching and aggregated cost functions is illustrated. The matching costs are obtained by Eq. 3.6 and the aggregated costs are obtained by Eq. 2.37. It can be observed that in both non-aggregated (left) and aggregated (right) cost functions (wrt. disparity), it is possible to estimate a disparity cost from its immediate neighbours by using linear interpolation. This situation is depicted in Fig. 3.3. The estimations which are performed within a local neighbourhood of the known cost values are acceptable. However, that is not guaranteed for the estimates of the cost values of the disparities which are relatively different from the disparities whose costs

[3] The actual assumption is based on irradiance of an image point. Colour intensity variation implies linearity on both irradiance and surface albedo.

are known. For instance, in Fig. 3.3, the cost estimate of the disparity, which is encircled by a blue circle, is poor; since, that disparity is relatively distant to the disparities whose costs are known along the disparity dimension and local linear behaviour within that disparity range is violated. Therefore, the estimation of the unknown costs should be performed in a local neighbourhood of the known costs in order to provide a reliable cost aggregation. In other words, the disparity hypotheses of the adjacent pixels should be close to each other so that unknown costs values always lie within the local neighbourhood of the known cost values. Actually, it is enough to have similar disparity hypotheses only for adjacent pixels which are color-wise similar; since, color-wise dissimilar pixels do not influence each other due to the edge-aware behaviour of the recursive filtering procedure given by 2.37. Therefore, the second condition, which forces colour-wise similar pixels in a local region to have similar disparity hypotheses, is required to practically ensure that the estimation of the unknown costs is always performed within a local neighbourhood of the known costs. This condition is assumed to hold; since, colour-wise similar pixels are assumed to come from the same object in a local region and for this reason they should have similar disparity hypotheses. This assumption is very similar to underlying assumption of region based methods [33, 84].

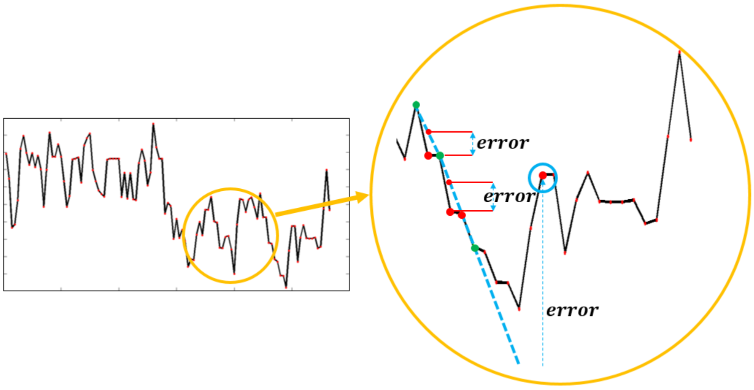


Figure 3.3: Cost estimation for 3 points marked by red from available cost values at the points marked by green. When linear interpolation is performed, the estimation error for the near points is small due to the linear behaviour of the function. Yet, the error significantly increases for the cost estimation at the distant point.

Apart from the complexity reduction challenge, this study is also motivated from the shortcomings of prior methods based on generating disparity candidate sets for pixels. The candidate set is generally formed via CTF approaches; since, CTF approaches can remove the complexity due to the disparity search range. As it is mentioned in Sec. 2.3.1.2, disparity estimate transfer between scales to generate disparity hypotheses is a problematic task due to propagation of erroneous estimates and it is dealt with local heuristic approaches focusing only this task for the sake of simplicity. It is observed that the pixels which have erroneous disparity estimates involve the occluded pixels. At this point, the question of a possible disparity transfer approach that can also handle occlusions is motivated the second part of the study. In this way, no additional computation would be required for occlusion handling and this gain from computation can be utilized by using more complex disparity transfer approaches without degrading the efficiency to improve the performance of CTF approaches. Therefore, the joint handling of disparity transfer and occlusion problems has been investigated and the solution has been sought by integrating depth super resolution approaches [47]. Finally, the problem is converted into an optimization problem and solved efficiently by DP.

3.2 Overview of the Method

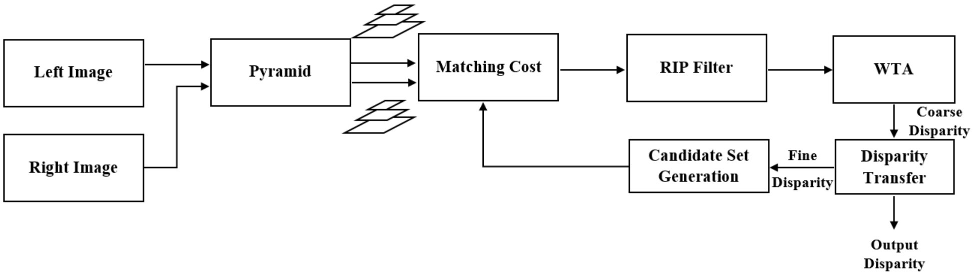


Figure 3.4: Flow diagram of the proposed algorithm. RIP Filter and Disparity Transfer blocks are the main blocks that are proposed in this study.

The proposed method follows a CTF scheme on image pyramids [46] constructed from left and right stereo image pairs. At each pyramid level, a sparse cost volume is formed according to the disparity hypotheses of each pixel, where the disparity hypotheses at the coarsest scale is the complete set of disparities in the disparity search range at that scale and for finer scales, it is a specified number of disparities that are picked according to the disparity estimates at the coarser scale. Values of the cost volume is determined according to a pixel-wise matching cost function. Once the sparse cost volume is specified, there are two main steps to generate disparity hypotheses for the finer scale. First one is cost aggregation via proposed efficient recursive edge-aware filtering for sparse cost volumes and the second one is proposed transfer of the reliable disparity estimates to finer scale. Following the cost aggregation, disparity estimation is performed by WTA approach. Then, estimates are transferred to the finer scale regarding to their reliabilities and hence, candidate set for each pixel is formed. The reliabilities are determined according to a novel strategy to be presented based on cost histogram. The flow diagram of the proposed stereo matching algorithm is provided in Fig. 3.4.

3.3 Image Pyramid Generation

The very first step of the overall method is the generation of the image pyramids from the left, $I_L[x,y]$, and the right, $I_R[x,y]$, input stereo image pairs, where an image, $I[x,y]$, is a function from $\mathrm{Z}^2 \rightarrow \mathrm{Z}^3$. I is a vector with components I^R, I^G, I^B. Each component corresponds to a color channel among red, R, green, G, and blue, B. Therefore, the value of $I[x,y]$ is a vector with color intensities at (x,y). An image pyramid, $\mathcal{P}$, is a set of images, as,

$$\mathcal{P} = \{I^{(0)}, I^{(1)}, \cdots, I^{(k)}, \cdots, I^{(n)}\},\ k \in \mathrm{Z}, \tag{3.1}$$

where the images, $\{I^{(0)}, I^{(1)}, \cdots, I^{(n)}\}$, are the downscaled images which are obtained from the original image, I. The set, $\mathcal{P}$, is formed hierarchically, as,

$$\begin{aligned} I^{(0)} &= I \\ I^{(1)} &= decimate[I^{(0)}, 2] \\ &\vdots \\ I^{(k)} &= decimate[I^{(k-1)}, 2] \\ &\vdots \\ I^{(n)} &= decimate[I^{(n-1)}, 2]\ , \end{aligned} \tag{3.2}$$

where $decimate[I, 2]$ performs low-pass filtering to the image, I, and downsamples the image, I, by 2 by rejecting odd coordinates. $I^{(0)}$ represents the finest scale image whereas $I^{(n)}$ represents the coarsest scale image. In $decimate[I, 2]$, low-pass filtering is achieved through convolving the image, I, with the Gaussian kernel, K, which is

$$K = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}. \tag{3.3}$$

Hence, the image, $I^{(k)}$, at the pyramid level, k, is obtained as,

$$I^{(k)}[x, y] = (I^{(k-1)} * K)[2\,x, 2\,y], \tag{3.4}$$

where $*$ denotes 2-D convolution. For a given maximum pyramid level, n, the left, $\mathcal{P}_L$, and the right, $\mathcal{P}_R$, image pyramids are constructed according to Eq. 3.4 by using the left, I_L and the right, I_R, input stereo image pairs. At each pyramid level, the two image pairs at that level are used to obtain a disparity map for that level.

3.4 Matching Cost Calculation

There exist many matching cost functions measuring the visual similarity of pixels between stereo images for a given disparity $d \in \mathrm{Z}$. An evaluation of the mostly utilized costs functions in terms of their computational complexities and matching reliabilities are provided in [32]. Among the methods investigated, *Census cost* [90] is reported to be one of the leading robust cost functions providing a reliable pixel similarity measure. Census cost is calculated after applying *Census Transform* (CT) to image pairs. Census cost is a non-parametric cost based on local order of intensities. Census Transform, $\underline{CT}[x, y]$, of a pixel at (x, y) is a bit string where each bit corresponds to a particular pixel in the local neighbourhood of the pixel of interest, $\mathcal{N}_{(x,y)}$. Given a reference image function

I, comparison of the intensity, $I[x,y]$, of the reference pixel at (x,y) and the intensity, $I[x_n,y_n]$ of the n^{th} neighbouring pixel which is at (x_n,y_n) is indicated by $CT[x,y,n]$. $CT[x,y,n]$ is the n^{th} bit of the bit string $\underline{CT}[x,y]$ and it is defined as,

$$CT[x,y,n] = \begin{cases} 1, & I[x,y] > I[x_n,y_n] \\ 0, & otherwise \end{cases} \;, \; (x_n,y_n) \in \mathcal{N}_{(x,y)} \;, \tag{3.5}$$

where n is a positive integer such that $1 \leqslant n \leqslant \| \mathcal{N}_{(x,y)} \|$. Here $\| \mathcal{N}_{(x,y)} \|$ denotes the number of pixels in the local neighbourhood, $\mathcal{N}_{(x,y)}$. In Fig. 3.5, numberings of the pixels in the local region of the pixel at (x,y) are given for 3 and 5 windows centred at (x,y). Different numbering conventions can be used; however, the relative location of the neighbouring pixel at (x_n,y_n) according to a pixel of interest, (x,y), should be the same for all pixels. In this way, $\underline{CT}[x,y]$ is an indicator of both intensity orderings and local spatial structure. Once $\underline{CT}[x,y]$ is obtained, matching cost of two pixels is determined by calculating Hamming distance[4] [32] of corresponding bit strings.

1	**2**	**3**
8	**(x,y)**	**4**
7	**6**	**5**

1	**2**	**3**	**4**	**5**
16	**17**	**18**	**19**	**6**
15	**24**	**(x,y)**	**20**	**7**
14	**23**	**22**	**21**	**8**
13	**12**	**11**	**10**	**9**

Figure 3.5: Numbering of the neighbouring pixels of (x,y) within windows of size 3 × 3 and 5 × 5.

In addition to Census cost, sum of absolute differences cost (SAD) is also utilized. It is very common and suggested to use weighted sum of more than one

[4] Hamming distance of two bit strings is the number of positions at which the corresponding bits are different.

matching cost and color based costs are encouraged [32]. Finally, the matching cost function, $MC[x,y,d]$, used in this study can be given for the case when the left image, $I_L[x,y]$ is taken as the reference image and the right image, $I_R[x,y]$, is taken as the matching image, as,

$$\begin{aligned} C_{SAD}[x,y,d] &= \sum_{i\in\{R,G,B\}} \left| I_L^i[x,y] - I_R^i[x-d,y] \right| \\ C_{CENSUS}[x,y,d] &= \parallel CT_L[x,y] - CT_R[x-d,y] \parallel_{HAMMING} \\ MC[x,y,d] &= min\left(\alpha\, C_{CENSUS}[x,y,d] + (1-\alpha)\, C_{SAD}[x,y,d], T\right), \end{aligned} \tag{3.6}$$

where SAD cost, C_{SAD}, is the sum of absolute differences of red, green and blue intensities. In order to reduce the sensitivity to the occlusion problem, truncated cost is used so that stronger smoothing is enforced to the occluded regions by setting an upper limit, T, to the costs. The matching cost is the weighted sum of SAD and Census cost and the weights are controlled by α. T and α are the external real valued parameters in $[0,1]$. The sparse matching cost volume can now be specified by calculating matching costs for every pixel, $\bar{p} = [x\ y]^T$, in the image at every disparity in its candidate set, $\mathcal{D}_{\bar{p}}$, as,

$$C[\bar{p},d] = \begin{cases} MC[\bar{p},d], & d \in \mathcal{D}_{\bar{p}} \\ not\ defined, & otherwise \end{cases} \tag{3.7}$$

where MC is given by Eq. 3.6 and formation of $\mathcal{D}_{\bar{p}}$ is explained in Sec. 3.6.2.3.

3.5 Efficient Sparse Cost Volume Filtering

In this section, a novel recursive edge-preserving filter to perform efficient cost aggregation in a sparse cost volume is presented. The recursive filtering structure given in Eq. 2.37 cannot be applicable to sparse cost volumes, since it is not guaranteed that the required cost information is available at preceding pixel location. In this study, it is proposed that efficient recursive filtering can be performed by predicting the required unknown data. Hence, the proposed filter is

$$C^f[\bar{p},d] = C[\bar{p},d] + \mu_{\bar{p},\bar{p}-\bar{r}}\, P(C^f_{\bar{p}-\bar{1}_r}, d)\ . \tag{3.8}$$

The filtered cost for assigning disparity d to pixel at location $\bar{p} = [xy]^T$ is determined by the weighted sum of the matching cost of that disparity assignment, $C[\bar{p}, d]$, and the predicted value, $P(C^f_{\bar{p}-\bar{r}}, d)$, of the filtered cost of that assignment to preceding pixel, $\bar{p} - \bar{r}$, in the corresponding recursion direction, $\bar{r}$. The adaptive filter coefficient, $\mu_{\bar{p},\bar{p}-\bar{r}}$, is

$$\mu_{\bar{p},\bar{q}} = \mathrm{e}^{-\dfrac{max\left\{|I^R[\bar{p}] - I^R[\bar{q}]|, |I^G[\bar{p}] - I^G[\bar{q}]|, |I^B[\bar{p}] - I^B[\bar{q}]|\right\}}{\sigma}} , \tag{3.9}$$

where σ is the decay control parameter that controls the smoothing based on the color similarity and I^R, I^G, I^B represent the red, green and blue color channels of the reference image, I. The selection of σ is based on Eq. 3.28 which is discussed in Sec. 3.5.3. The prediction function, $P(\cdot)$, can be a linear interpolation operation, as,

$$P(C_{\bar{p}}, d) = (d - d_1)\,\frac{C[\bar{p}, d_1] - C[\bar{p}, d_2]}{d_1 - d_2} + C[\bar{p}, d_1],\ \{d_1, d_2\} \in \mathcal{D}_{\bar{p}}\ , \tag{3.10}$$

or nearest neighbour interpolation, as,

$$P(C_{\bar{p}}, d) = C[\bar{p}, d^*],\ d^* = \underset{k \in \mathcal{D}_{\bar{p}}}{\arg\min}\, |d - k|\ , \tag{3.11}$$

where d_1 and d_2 correspond to the two closest disparities to the disparity d and $\mathcal{D}_{\bar{p}}$ is the disparity candidate set of the pixel $\bar{p}$. Prediction function, $P(\cdot, \cdot)$, can be viewed as a function of 1-D sparse cost function of a pixel and a target disparity, d. The value of this function is the estimate of the value of the given 1-D sparse cost function at d. In other words, $P(\cdot, \cdot)$ is the operation of estimation of the unknown cost values from known cost values which are present in 1-D sparse cost function. What is meant by 1-D sparse cost function of a pixel is actually the indication that only the costs of the disparities in the candidate set, $\mathcal{D}_{\bar{p}}$, are known and the 1-D sparse cost function, $C_{\bar{p}}$, of a pixel, $\bar{p}$, in Eq. 3.8, is

$$C_{\bar{p}}[d] = \begin{cases} C[\bar{p}, d], & d \in \mathcal{D}_{\bar{p}} \\ not\ defined, & otherwise \end{cases} , \tag{3.12}$$

where $C[\bar{p}, d]$ is the matching costs given by Eq. 3.7.

The filter given by Eq. 3.8 is named as *Recursive Information Prediction* (RIP) filter. In order to perform cost aggregation over a sparse cost volume, RIP filter is applied along the horizontal and vertical directions[5] sequentially similar to 2-pass 1-D filtering introduced in Sec. 2.3.1.1. The difference is that the costs of the each disparity in the candidate set of a pixel should be processed by Eq. 3.8 prior to proceeding to the next pixel; since, those costs are utilized to estimate unknown cost value while processing the proceeding pixel. That is, horizontal passes are performed on matching cost volume, C, given by Eq. 3.7 and at each pixel $\bar{p}$, Eq. 3.8 is applied for each disparity candidate, $d \in \mathcal{D}_{\bar{p}}$, as,

$$\begin{aligned} \underset{L\to R}{C}[\bar{p}, d] &= C[\bar{p}, d] + \mu_{\bar{p},\bar{p}-\bar{r}_h}\, P(\underset{L\to R}{C_{\bar{p}-\bar{r}_h}}, d) \\ \underset{R\to L}{C}[\bar{p}, d] &= C[\bar{p}, d] + \mu_{\bar{p},\bar{p}+\bar{r}_h}\, P(\underset{R\to L}{C_{\bar{p}+\bar{r}_h}}, d) \\ \acute{C}[\bar{p}, d] &= \underset{L\to R}{C}[\bar{p}, d] + \underset{R\to L}{C}[\bar{p}, d] - C[\bar{p}, d]\ , \end{aligned} \tag{3.13}$$

where $\bar{p} = [x\ y]^T$ and $\bar{r}_h = [1\ 0]^T$. After the horizontal passes, vertical passes are performed on updated cost volume, $\acute{C}$, as,

$$\begin{aligned} \underset{T\to B}{\acute{C}}[\bar{p}, d] &= \acute{C}[\bar{p}, d] + \mu_{\bar{p},\bar{p}-\bar{r}_v}\, P(\underset{T\to B}{\acute{C}_{\bar{p}-\bar{r}_v}}, d) \\ \underset{B\to T}{\acute{C}}[\bar{p}, d] &= \acute{C}[\bar{p}, d] + \mu_{\bar{p},\bar{p}+\bar{r}_v}\, P(\underset{B\to T}{\acute{C}_{\bar{p}+\bar{r}_v}}, d) \\ \check{C}[\bar{p}, d] &= \underset{T\to B}{\acute{C}}[\bar{p}, d] + \underset{B\to T}{\acute{C}}[\bar{p}, d] - \acute{C}[\bar{p}, d]\ , \end{aligned} \tag{3.14}$$

where similarly $\bar{r}_v = [1\ 0]^T$. Finally, the aggregated costs, C^f, are obtained after normalization, as,

$$\begin{aligned} C^f[\bar{p}, d] &= \frac{\check{C}[\bar{p}, d]}{\eta[\bar{p}, d]} \\ \eta &= \mathcal{F}(1s)\ , \end{aligned} \tag{3.15}$$

where $1s$ is the dummy data consisting of ones and $\mathcal{F}$ is the filtering process given by Eqs. 3.13 and 3.14 such that $\mathcal{F}(C) = \check{C}$. Accumulated values in η correspond to sum of effective weights of the supporting pixels, thanks to ones being identity element of multiplication operation. The prediction function, $P(\cdot, \cdot)$, which appears in Eqs. 3.13 and 3.14 is given by Eqs. 3.10 and 3.11.The explicit flow of the proposed filtering and corresponding cost aggregation is provided as a pseudo code in Algorithm 1 and 2, 3.

[5] These directions are according to image coordinates, i.e., horizontal and vertical directions are along x and y coordinates, respectively.

Algorithm 1 Proposed Filtering

```
Algorithm RIPFilter($C$,$\mu$,$\mathcal{D}$,$r$,$p$)
Input:
$C$: sparse cost slice along a direction of length L
$\mu$: set of all filter coefficients along a direction
$\mathcal{D}$: set of all disparity candidate sets along a direction
$r$: 2-D vector to reach preceding location in recursion direction
$p$: 2-D vector indicating initial pixel location
Output:
$C^f$: filtered sparse cost slice along the given direction
begin
for each disparity $d$ in candidate set $D_p$ do
    initialize $C^f$: $C^f[p,d]$ $C[p,d]$
end for
for $q \leftarrow (p+r)$ to $(p+L\,r)$ do
    for each disparity $d$ in candidate set $D_q$ do
        $C^f[q,d] \leftarrow C[q,d] + \mu_{q,q-r}\,P(C^f_{q-r},d)$   ▷ $P$ is given by Eqs. 3.10 and 3.11
    end for
end for
end
```

Algorithm 2 Proposed Cost Aggregation

```
Algorithm CostAggregation(C,I,𝒟,σ)
Input:
$C$: sparse cost volume
$I$: the reference image
$\mathcal{D}$: set of all disparity candidate sets
$\sigma$: decay control parameter for filter coefficients
Output:
$C^f$: filtered sparse cost volume
begin
initialize horizontal and vertical direction vectors:
$r_h \leftarrow (1, 0)$
$r_v \leftarrow (0, 1)$
for each disparity pixel $p$ in $I$ do
    initialize buffers $\mu_{L2R}$, $\mu_{R2L}$, $\mu_{T2B}$, $\mu_{B2T}$ to store all filter coefficients along the directions left-to-right, right-to-left, top-to-bottom and bottom-to-top, respectively:
    $\mu_{L2R}[p, p - r_h] \leftarrow \mu_{p,p-r_h}$
    $\mu_{R2L}[p, p + r_h] \leftarrow \mu_{p,p+r_h}$
    $\mu_{T2B}[p, p - r_v] \leftarrow \mu_{p,p-r_v}$
    $\mu_{B2T}[p, p + r_v] \leftarrow \mu_{p,p+r_v}$
                                        ▷ $\mu_{p,q}$ is given by Eq. 3.9
end for
```

Algorithm 3 Proposed Cost Aggregation Continued

```
for each disparity cost slice along horizontal direction $C_h$ in $C$ do
    $p \leftarrow first\ pixel\ of\ C_h$
    $p \leftarrow\ last\ pixel\ of\ C_h$
    $\mu^h_{L2R} \leftarrow corresponding\ horizontal\ row\ of\ \mu_{L2R}$
    $\mu^h_{R2L} \leftarrow corresponding\ horizontal\ row\ of\ \mu_{R2L}$
    $\mathcal{D}^h \leftarrow set\ of\ all\ candidate\ sets\ along\ corresponding\ horizontal\ direction$
    $C^{L2R}_h \leftarrow RIPFilter(C_h, \mu^h_{L2R}, \mathcal{D}^h, r_h, p)$
    $C^{R2L}_h \leftarrow RIPFilter(C_h, \mu^h_{R2L}, \mathcal{D}^h, -r_h, q)$
    $C^f_h \leftarrow C^{L2R}_h + C^{R2L}_h - C_h$
end for
                    ▷ at this point $C^h$ denotes filtered $C$ in horizontal directions
for each disparity cost slice along vertical direction $C_v$ in $C^h$ do
    $p \leftarrow first\ pixel\ of\ C_v$
    $p \leftarrow\ last\ pixel\ of\ C_v$
    $\mu^v_{T2B} \leftarrow corresponding\ vertical\ column\ of\ \mu_{T2B}$
    $\mu^v_{B2T} \leftarrow corresponding\ vertical\ column\ of\ \mu_{B2T}$
    $\mathcal{D}^h \leftarrow set\ of\ all\ candidate\ sets\ along\ corresponding\ vertical\ direction$
    $C^{T2B}_v \leftarrow RIPFilter(C_v, \mu^v_{T2B}, \mathcal{D}^h, r_v, p)$
    $C^{B2T}_v \leftarrow RIPFilter(C_v, \mu^v_{B2T}, \mathcal{D}^h, -r_v, q)$
    $C^f \leftarrow C^{T2B}_v + C^{B2T}_v - C_v$
end for
apply cost aggregation process described above to a dummy data consisting
    of 1s and perform normalization given by Eq. 3.15
end
```

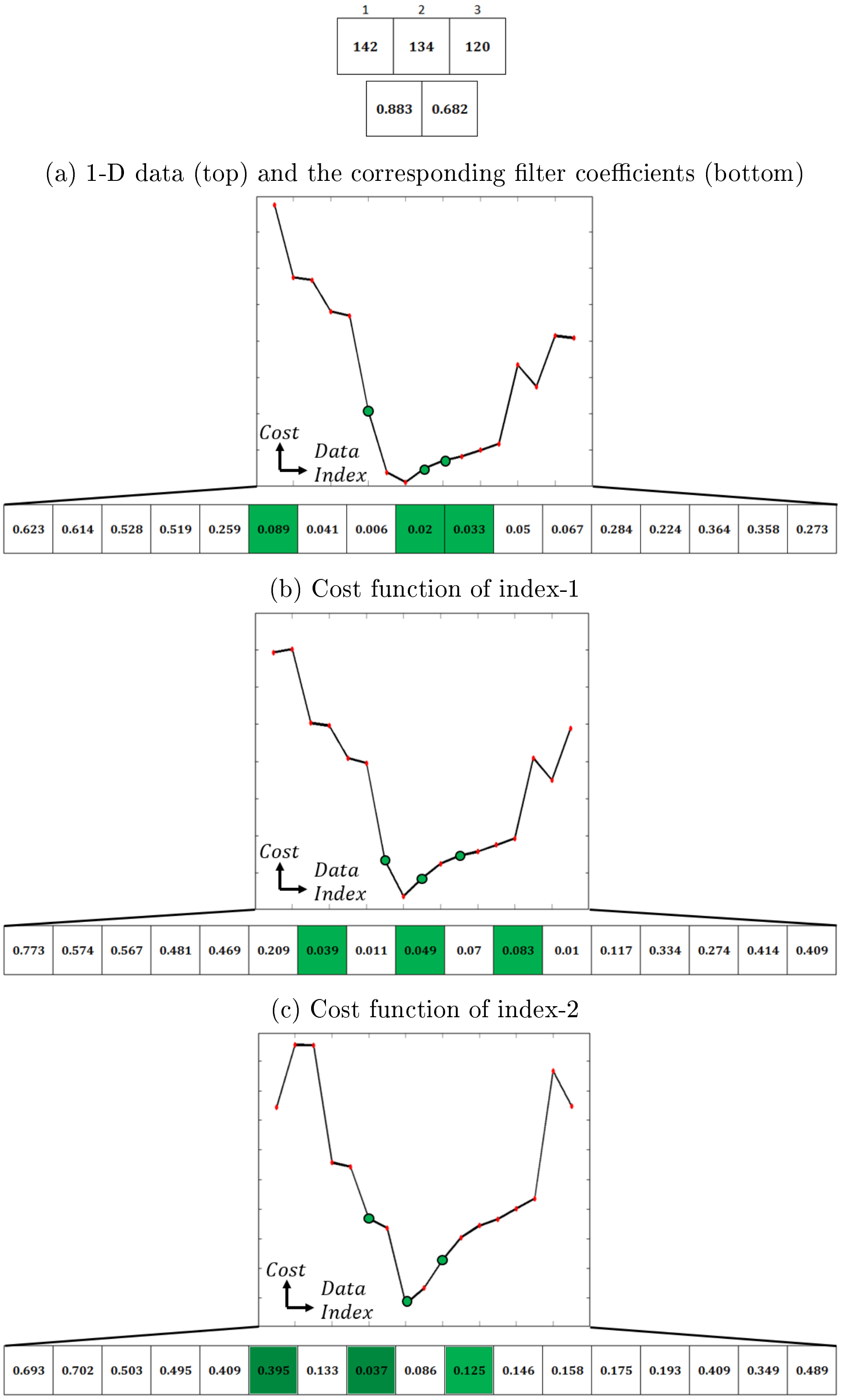
The proposed filtering allows efficient[6] cost aggregation over sparse cost volumes. The efficiency makes sense provided that the prediction step does not introduce additional complexity factor and can estimate unknown cost values approximate to actual ones. This can be satisfied as long as the two conditions described in the previous Sec. 3.1 hold. The first condition which ensures piecewise linear behaviour of the cost functions enables estimation of unknown costs efficiently by linear interpolation. Moreover, the second condition provides reliable cost aggregation by guaranteeing the propagation of reliable cost estimates among color-wise similar pixels only and preventing unreliable estimates from influencing the aggregation. In order to prove reliability of the proposed cost aggregation and to better explain the steps in the pseudo code, a simple cost aggregation example is provided in the next part of this section. Additionally, the prediction step is explained and its effects on complexity is discussed in the following parts of this section.

The edge-preserving behaviours of such recursive filters have already been analysed in Sec. 2.3.1.1. As it is mentioned, the color similarity adaptive filter coefficient, μ, controls influence passing between pixels so that while colour-wise similar pixels support each other, the influence of dissimilar pixels are avoided. μ is controlled by a decay parameter σ which determines the value of color intensity difference that should prevent information passing. In previous works [16, 60], this parameter is tuned empirically and fixed. However, it is observed that this parameter should be adaptive to images to be processed due to different contrast characteristics of the images. This issue is addressed in this work and an adaptive selection of σ parameter is proposed. This strategy is explained in the following parts of this section.

3.5.1 Example of a Simple Cost Aggregation

In this section, the proposed cost aggregation is applied to a small part of a data to clarify the method and to support its reliable cost aggregation. The cost aggregation is performed on the 1-D data given in Fig. 3.6. The image intensity

[6] The efficiency in this context is fast and reliable cost aggregation.



1	2	3
142	134	120

0.883	0.682

(a) 1-D data (top) and the corresponding filter coefficients (bottom)

0.623	0.614	0.528	0.519	0.259	0.089	0.041	0.006	0.02	0.033	0.05	0.067	0.284	0.224	0.364	0.358	0.273

(b) Cost function of index-1

0.773	0.574	0.567	0.481	0.469	0.209	0.039	0.011	0.049	0.07	0.083	0.01	0.117	0.334	0.274	0.414	0.409

(c) Cost function of index-2

0.693	0.702	0.503	0.495	0.409	0.395	0.133	0.037	0.086	0.125	0.146	0.158	0.175	0.193	0.409	0.349	0.489

(d) Cost function of index-3

Figure 3.6: 1-D data for the cost aggregation example via proposed RIP Filtering. The points marked by green are known and cost aggregation is to be performed by passing the data from left to right.

values and the corresponding filter coefficients are provided in Fig. 3.6a. The sparse matching cost functions at the corresponding locations are illustrated in Figs. 3.6b, 3.6c and 3.6d. The information spotted by the green dots are known for three different indices. The cost aggregation is performed from left-to-right.

The aggregated costs for the first index are the matching costs at the corresponding locations, as,

$$\begin{aligned} C^f[1,6] &= C[1,6] = 0.089 \ , \\ C^f[1,9] &= C[1,9] = 0.02 \ , \\ C^f[1,10] &= C[1,9] = 0.033 \ . \end{aligned} \tag{3.16}$$

Now, each matching cost of the second index is filtered according to Eq. 3.8. It is important to note that, prior to proceeding to the next index, third one, all the costs at the second index should be filtered; since, those costs is to be used to predict required unknown information. Thus, the aggregated cost of the second index is

$$C^f[2,7] = C[2,7] + \mu_{2,1}\, P(C^f_1, 7) \ . \tag{3.17}$$

Since the previous index does not have the required cost information, $C^f[1,7]$, this information is predicted from already available information at the previous index. The nearest available information to $C^f[1,7]$ is $C^f[1,6]$ and $C^f[1,9]$. Assuming local linear behaviour, the required information is estimated via linear interpolation by using $C^f[1,6]$ and $C^f[1,9]$, as,

$$\begin{aligned} C^f_{apprx}[1,7] &= (7-6)\,\frac{C^f[1,6] - C^f[1,9]}{6-9} + C^f[1,6] \\ C^f_{apprx}[1,7] &= 0.066 \ . \end{aligned} \tag{3.18}$$

Fig. 3.7 depicts the prediction of the unknown cost value to be used in cost aggregation. Now, this estimate is used to calculate $C^f[2,7]$ as:

$$\begin{aligned} C^f[2,7] &= C[2,7] + \mu_{2,1}\, P(C^f_1, 7) \\ C^f[2,7] &= 0.039 + 0.883 \cdot 0.066 = 0.0973 \ . \end{aligned} \tag{3.19}$$

Similarly, $C^f[2,9]$ and $C^f[2,11]$ are determined. For $C^f[2,9]$, the aggregated cost value is directly available and for $C^f[2,11]$, the aggregated cost value of

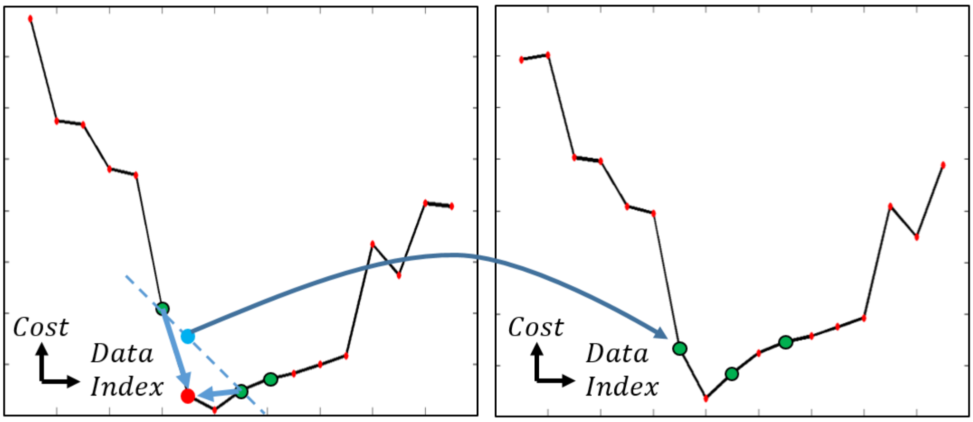


Figure 3.7: Estimation of the unknown cost value in order to perform cost aggregation. The aggregated cost value of the index-2 (right) is determined by using the aggregated costs of the index-1 (left). For index-2, in order to calculate the aggregated cost value at 7, the aggregated cost value of the index-1 at 7 is required. Yet, it is not available. It is estimated by utilizing the nearest two available cost value and that estimate is contributed to the cost aggregation for index-2.

$C^f[1,9]$ and $C^f[1,10]$ is used to estimate $C^f[1,11]$. Consequently,

$$\begin{aligned} C^f[2,9] &= C[2,9] + \mu_{2,1}\, P(C_1^f, 9) \\ C^f[2,9] &= 0.049 + 0.883 \cdot 0.02 = 0.0667 \ , \end{aligned} \tag{3.20}$$

$$\begin{aligned} C^f[2,11] &= C[2,11] + \mu_{2,1}\, P(C_1^f, 11) \\ P(C_1^f, 11) &= (11-9)\,\frac{0.02 - 0.033}{9-10} + 0.02 = 0.0460 \\ C^f[2,11] &= 0.083 + 0.883 \cdot 0.0460 = 0.1236 \ . \end{aligned} \tag{3.21}$$

Once all the available costs are processed, the costs of the next index can be processed. Applying the RIP filter, the aggregated costs of the third index is determined as:

$$\begin{aligned} C^f[3,6] &= C[3,6] + \mu_{3,2}\, P(C_2^f, 6) \\ P(C_2^f, 6) &= (6-7) \cdot \frac{0.0973 - 0.0667}{7-9} + 0.0973 = 0.1126 \\ C^f[3,6] &= 0.395 + 0.682 \cdot 0.1126 = 0.4718 \ , \end{aligned} \tag{3.22}$$

$$\begin{aligned} C^f[3,8] &= C[3,8] + \mu_{3,2}\, P(C_2^f, 8) \\ P(C_2^f, 8) &= (8-7) \cdot \frac{0.0973 - 0.0667}{7-9} + 0.0973 = 0.0820 \\ C^f[3,8] &= 0.037 + 0.682 \cdot 0.0820 = 0.0929 \ , \end{aligned} \tag{3.23}$$

$$\begin{aligned} C^f[3,10] &= C[3,10] + \mu_{3,2}\,P(C^f_2,10) \\ P(C^f_2,10) &= (10-9)\cdot\frac{0.0667-0.1236}{9-11} + 0.0667 = 0.0951 \\ C^f[3,8] &= 0.125 + 0.682\cdot 0.0951 = 0.1899\ . \end{aligned} \tag{3.24}$$

After aggregating the costs, the normalization should be performed. The normalization coefficients for the indices are 1, $(1+\mu_{2,1})$ and $(1+\mu_{2,1}\,\mu_{3,2}+\mu_{3,2})$ respectively. After the normalization , the aggregated costs of the third index is plotted in Fig. 3.8 against full cost aggregation using recursive EAF. It can be observed that the aggregation via RIP Filtering provides good approximation to aggregated costs as long as its requirements are satisfied.

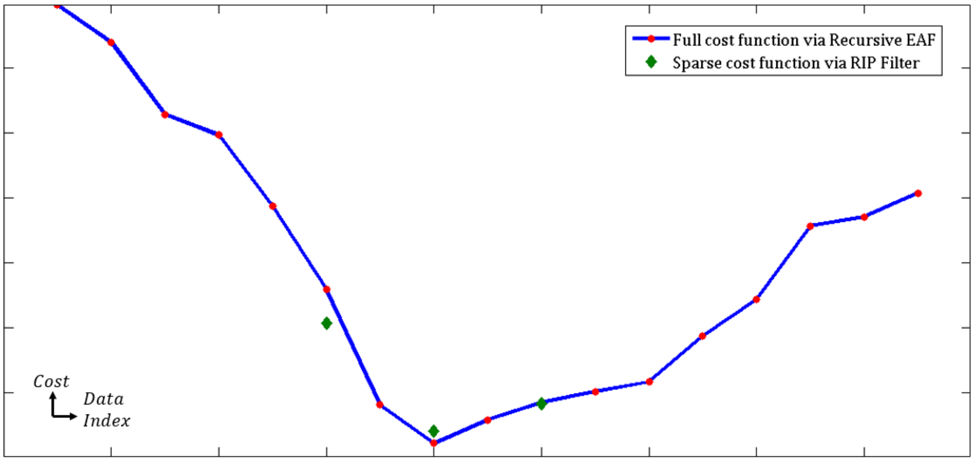


Figure 3.8: Comparison of the aggregated cost values at index-3.

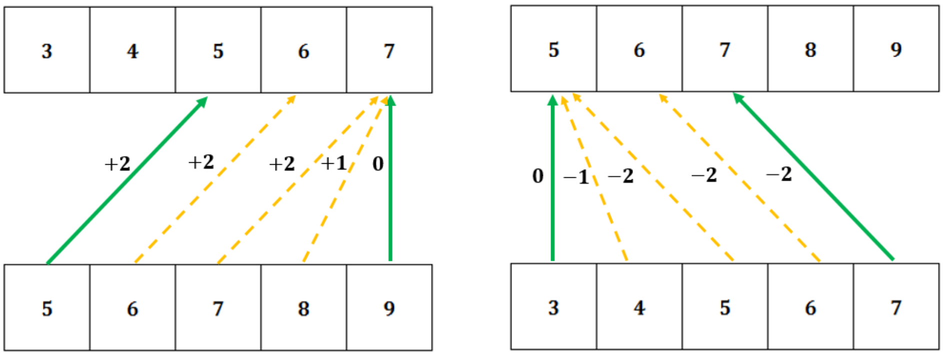
3.5.2 Prediction of Cost Values, $\mathbf{P(C^f_p, d)}$

Assuming that the conditions given in Sec. 3.1 are satisfied, an unknown cost value for a disparity can be estimated well by linear interpolation of the known cost values of the two closest disparities as given by Eq. 3.10. This is supported in the previous part by an example. Estimation of unknown costs introduces additional complexity to the recursive filtering process. The complexity comes from two aspects: linear interpolation and search for nearest two neighbours. The former is just some fixed additional operations and can be ignored. It does not depend on either image size or disparity search range. However, the latter comes up with an additional factor of complexity. If the number disparities in the candidate set is D_C, then filtering complexity becomes $\mathcal{O}(N \cdot D_C \cdot \log(D_C))$, where N is the image size. Factor $\log(D_C)$ comes from the search for the closest two disparities where the costs are available. Actually, the operations performed during search are elementary operations and take quite less time than operations, such as multiplication. Therefore, the real effect of that complexity is not relatively high when considered together with other factors; however, it is still a factor to be considered.

The complexity factor coming from nearest neighbour search can be eliminated provided that some constraints are imposed on disparity candidate set. If a disparity candidate set is an ordered set and two consecutive disparities in the set correspond to two consecutive disparities in the disparity domain, then the complexity due to search can be eliminated. Explicitly, the constraints for a disparity candidate set, $\mathcal{D}_{\bar{p}}$, are

$$\mathcal{D}_{\bar{p}} = \{d_i : d_{i+1} - d_i = T_d\},\ i \in \mathrm{Z}^+ \ , \tag{3.25}$$

where T_d denotes the sampling interval of disparity space and it can be considered as the resolution of the disparity domain. According to these constraints, if the sampling interval is T_d is 1, then the disparity candidate set $\{5, 6, 7, 8, 9\}$ satisfies the constraints, whereas the set $\{5, 8, 9, 11, 12\}$ does not.



(a) Offsets when offset of the first element is positive

(b) Offsets when offset of the last element is negative

Figure 3.9: Determination of nearest neighbours of the elements in the set. Two cases are possible: (a) Starting from the last element, the offsets are assigned increasingly up to offset of the first element. (b) Starting from the first element, the offsets are assigned decreasingly down to offset of the last element.

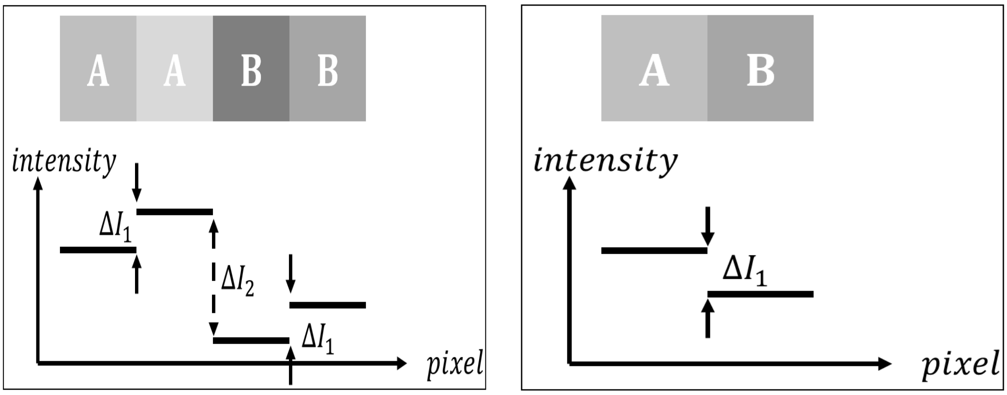
Given that two sets obey the constraints in Eq. 3.25, then the nearest neighbours of the disparities in one set in the other set can be found by performing only two searches which are for the first and the last disparity in the set. Once the nearest neighbours of the first and the last disparity in the set are known, then the nearest neighbours of the rest can be easily determined with fixed number of operations as illustrated in Fig. 3.9. In this way, the complexity of the nearest neighbour search results in an additional complexity.

With this new set formation, the linear interpolation step is not required for most of the cases, since color-wise similar pixels have similar disparity candidate sets. Therefore, for further reduction on complexity, nearest neighbour interpolation can be performed instead of linear interpolation as given by Eq. 3.11.

3.5.3 Adaptive Selection of Decay Control Parameter, σ

In Eq. 2.36, σ controls the decay rate of the weighting function, μ. In other words, edge-preserving behaviour is dependent on it. In previous works, this parameter is fixed. However, gradients on object boundaries generally differ from image to image due to many factors such as illumination, characteristic of the scene, sensors of camera etc. An intensity difference which is considered as insignificant may become significant for a low contrast image as shown in Fig. 3.10. Parts of two intensity images are provided in In Figs. 3.10a and 3.10a, where two different objects are indicated by A and B. In Fig. 3.10a, an intensity difference of ΔI_1 should be considered as insignificant; since, it is an intensity variation within the same object. On the other hand, the same intensity difference corresponds to an edge between the two objects for the image in Fig. 3.10b and it should be considered as significant in this case. Thus, a parameter setting according to high contrast image results over-smoothing in low contrast image. An example situation is depicted in Fig. 3.11, where two disparity estimation results are provided under fixed σ. For the image on the left, the parameter works fine while for the other image, it causes over-smoothing. To handle this problem, an automatic selection of σ parameter is proposed.

It is assumed that as long as there are different objects in the scene, there exist edges and these edges constitute a small fraction of an image. In other words,



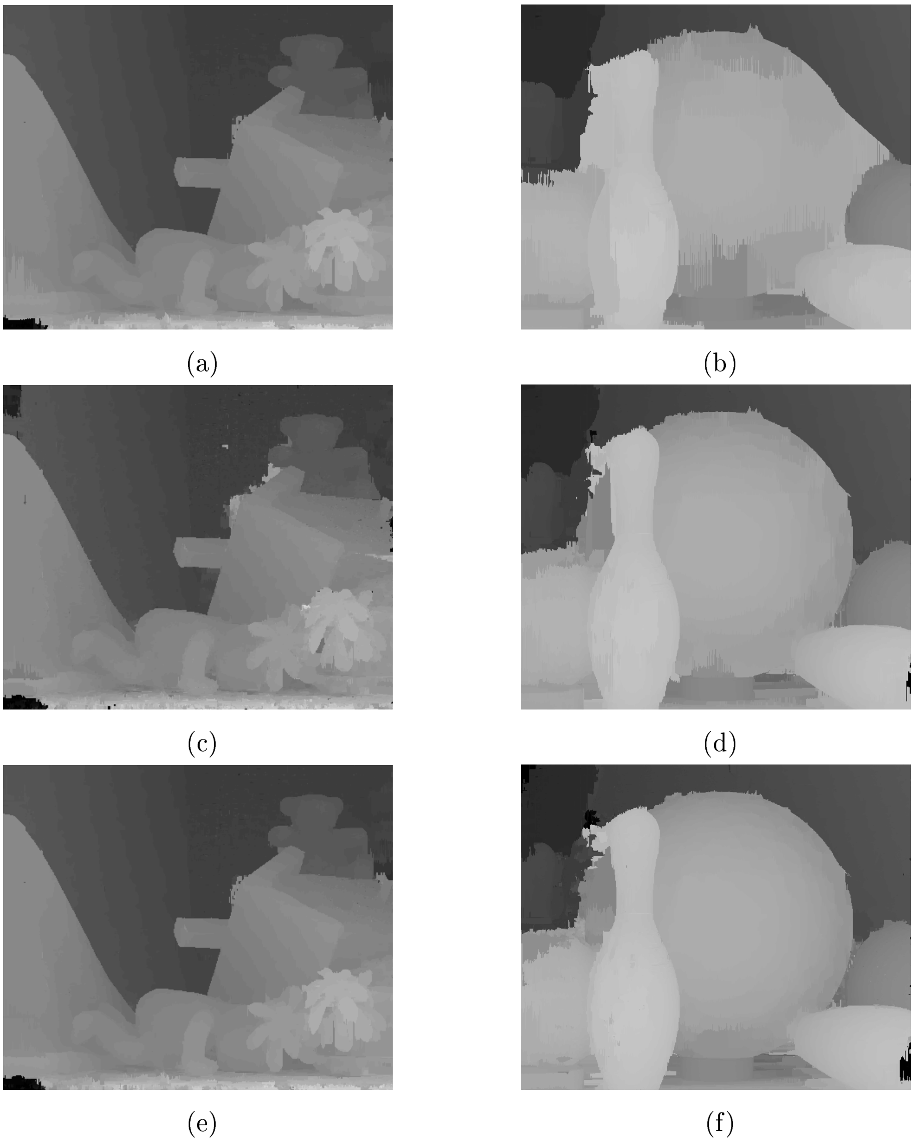
(a) (b)

Figure 3.10: Different significance of the same intensity difference for different contrast images.

the portion of the homogeneous regions in an image is much larger than the portion of the discontinuities. This assumption is verified via experiments with approximately 30 images from different indoor scenes. The edges are extracted with *Canny Edge Detector* [14] and the proportion of the pixels on the edges are analysed. It is observed that approximately 10 percent of the pixels lie on an edge in general. In order to utilize this information, another reasonable assumption, which states that pixels on the edges have larger gradient magnitudes, is made. If the minimum gradient magnitude that an edge pixel can have is determined approximately, then decay control parameter can be determined accordingly: The weight function, μ, given by Eq. 2.36 should get very small values at the minimum gradient magnitude that an edge pixel can have in order to prevent influence of a pixel from passing beyond an edge. Hence, it is proposed that σ can be selected adaptively by analysing the gradient magnitudes in an image in the following manner.

The gradient magnitudes in the image, $| \triangledown I|$, are calculated as,

$$\begin{aligned} | \triangledown I[x,y]| = max\{&|I[x,y] - I[x-1,y]|, \\ &|I[x,y] - I[x+1,y]|, \\ &|I[x,y] - I[x,y-1]|, \\ &|I[x,y] - I[x,y+1]|\} \ . \end{aligned} \tag{3.26}$$



(a) (b)

(c) (d)

(e) (f)

Figure 3.11: Illustration of sensitivity to smoothness parameter. Top row: The parameter is set to the value at which the estimation performance is the best for the left scene. Yet, this parameter is not suitable for the right scene. It causes over-smoothing. Middle row: The parameter is set according to right scene. In this case, the estimation performance of the left scene degrades. Bottom row: The parameter is set automatically by the proposed approach. Proposed approach enables adaptation to the scene.

Then, a threshold, th, is determined to satisfy,

$$th : \frac{\sum\limits_{(x,y):|\triangledown I[x,y]|\geqslant th} 1}{\sum\limits_{\forall(x,y)\in I} 1} \leqslant T_{edge} \ . \tag{3.27}$$

In this study, T_{edge} is selected as 0.15 based on the experiments that are conducted on approximately 30 indoor scenes. The underlying reason of determining the threshold, th, is to estimate the possible minimum gradient magnitude on an edge, approximately. The gradient magnitudes that are greater than this threshold are considered to be present in an edge transition. The value of the weighting function ,μ, given by Eq. 2.36 should be zero at the gradient magnitudes that the edge pixels have. In this way, edge-aware cost aggregation can be performed by preventing two pixels from different objects from influencing each other during cost aggregation. Hence, σ is selected such that the value of the weighting function ,$\mu = \mathrm{e}^{-\frac{th}{\sigma}}$, becomes almost zero at the threshold, th, as,

$$\sigma = \frac{th}{3} \ , \tag{3.28}$$

since, $\mathrm{e}^{-3} \approx 0$.

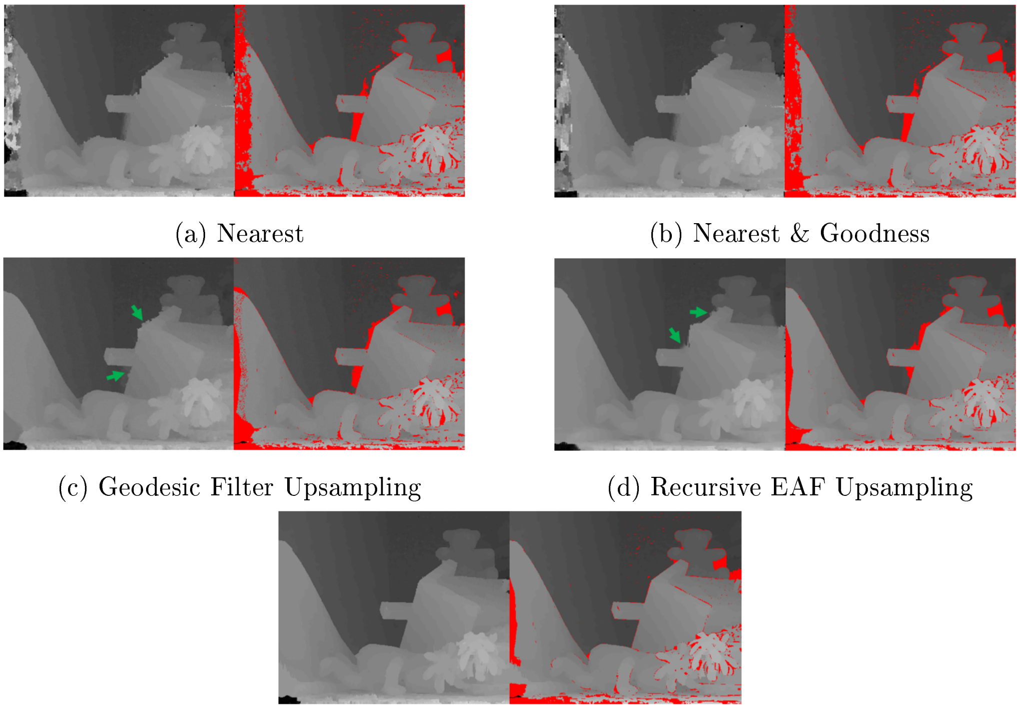
In Fig. 3.11, the effect of adaptive selection of decay control parameter is presented. It should be emphasized that this adaptive selection of decay control parameter is proposed according to indoor scenes. The threshold presented in Eq. 3.27 is determined according to the experiments that are conducted on various indoor scenes. Moreover, the assumptions that are made to propose an automatic selection of the parameter are based on indoor scenes.

3.6 Disparity Transfer and Asymmetric Occlusion Handling

This section proposes a novel method to transfer disparity estimates from coarser scales to finer scales in a CTF approach. The disparity candidate sets of the pixels in a fine scale image are formed according to the disparity estimations of the coarser scale. As it is explained in the previous chapter in Sec. 2.3.1.2, using the right information from coarser scale plays an important role in the performance. Poor disparity estimates while forming the disparity candidate sets cause propagation of erroneous disparity estimates between scales and this results in problems like fine structure loss or unsatisfactory results at object boundaries. In Fig. 3.12, sample disparity estimation results of different disparity estimate transfer approaches are presented. The cost aggregation is performed by the proposed cost aggregation method. The topmost two disparity maps given by

Figs. 3.12a and 3.14b are the results of commonly utilized nearest neighbour methods [36, 37, 66]. In these methods, the coarse disparity map is upsampled to form a fine disparity map and the fine disparity map is utilized to generate disparity hypotheses at the fine scale. The upsampling is performed via nearest neighbour approach within a window. In Fig. 3.12a, the disparity estimation result of color-wise nearest neighbour approach is illustrated. The nearest neighbour is determined according to color-wise similarity. In Fig. 3.14b, the disparity estimation result of matching cost based nearest neighbour approach [66] is illustrated. In that approach, the costs of the disparity estimates at the coarse scale is utilized while determining the nearest neighbour of a pixel. In these window based approaches, the size of the window is kept small for the sake of less computational load. However, small windows prevent utilizing reliable disparity estimates of distant pixels; and this results in poor estimates especially at object boundaries as shown in Fig. 3.12. The other three disparity maps given by Figs. 3.12c, 3.12d and 3.12e are the results of the proposed reliable disparity estimate transfer approaches. These approaches are non-local approaches and it can be observed they improve estimations at the object boundaries.

In order to transfer disparity estimates according to reliability, the reliability of a disparity estimate should be determined first and the disparity candidate sets for the pixels in the fine image should be constructed based on the reliable estimates. Best clues to decide the reliabilities of the disparity estimates are the costs associated with the estimates and the geometric constraints utilized in occlusion handling [66]. In this study, the reliability is determined regarding to costs corresponding to disparity estimates; since, it is considered to be more general. In other words, it is assumed that unreliable estimates based on costs associated with them contains the estimates violating geometric constraints (visibility, uniqueness) as well. This assumption implies that the detection of unreliable estimates inherently detects the occluded pixels. Hence, the problems of occlusion handling and reliable disparity estimate transfer can be coupled. The occluded regions can be filled while transferring the disparity estimates from coarser scale. This leads to reductions on computational load; since, the disparity transfer, which is the necessary of a CTF approach, handles the occluded



(a) Nearest

(b) Nearest & Goodness

(c) Geodesic Filter Upsampling

(d) Recursive EAF Upsampling

(e) Proposed Disparity Transfer

Figure 3.12: Comparison of methods for disparity estimate transfer (upsampling) from coarse scale to fine scale. The erroneous estimates are indicated by red. (a) and (b) are local approaches that work in a window. (c) and (d) are proposed DSR based disparity transfer. Except for bleeding artefacts pointed by green arrows, they perform quiet well as transferring the disparity estimates and filling the occlusions. (e) is the proposed optimization based transfer method. It improves the results at object boundaries and handles occlusions better.

regions. Thanks to this gain from computation, a bit more complex approach than the existing solutions is utilized to improve performance due to disparity transfer step. The proposed disparity transfer method is explained in two parts as determining the reliability of the estimates and disparity transfer.

3.6.1 Assigning Reliability to Disparity Estimates

The purpose of this step is to favour the disparity estimates that can be trusted while deciding the disparity hypotheses of the pixels in the finer scale. In this way, disparity candidate set formation in the finer scale disregards the unreliable disparity estimates at the coarse scale. The costs associated with the disparity estimates determine whether the estimates are decent. Associated cost, $C^{min}[\bar{p}]$, at a pixel location $\bar{p}$ is,

$$C^{min}[\bar{p}] = \min_{d \in \mathcal{D}_{\bar{p}}} C^f[\bar{p}, d] \ , \tag{3.29}$$

which corresponds to the minimum cost among disparity candidates and it is actually the cost of the disparity assignment after cost aggregation; since, disparity assignment is performed via WTA optimization. $C^f[\bar{p}, \cdot]$ in Eq. 3.29 is obtained by 3.15 and it holds the aggregated costs for disparity candidates of the pixels, $\mathcal{D}_{\bar{p}}$.

Costs of the disparity assignments are already considered in the literature [2,57, 67,84,92] to detect unreliable pixels for occlusion handling purposes. A hard threshold for costs is used to determine whether a disparity estimate at a pixel is reliable as in Eqs. 2.39 and 2.40. In this study, such an approach is not favoured; since, it imposes hard constraint on transfer of disparities to finer scale, and that can be problematic. Thresholding classifies pixels as reliable and unreliable. Then, only the disparities of the reliable pixels are used as reference disparities in the finer scale. However, pixels with costs close to edge of the threshold can be misclassified and also the cost threshold might differ from image to image. Therefore, it is proposed to impose reliability as a soft constraint on disparity transfer to finer scale. For this purpose, a confidence cost indicating how decent the disparity estimate is assigned to each pixel with lower costs meaning more reliable estimates.

It is assumed that the costs of the disparity estimates, C^{min}, come from a bimodal distribution, $g(x)$. The underlying reason behind this assumption is that in general, the pixels are assigned to correct disparity values and the exceptions are the occluded regions and the assignments owing to erroneous propagation of the disparity information from coarser scales. The costs of such regions should be high while the costs of correct assignment are being low, provided that a reliable matching cost and cost aggregation method is used. Moreover, it is assumed that the bimodal distribution of the costs, $g(x)$, can be approximated by mixture of two Gaussian distributions, $\mathcal{N}(\mu, \sigma)$, with mean, μ, and standard deviation, σ, as,

$$g(x) = \alpha \mathcal{N}(\mu_1, \sigma_1)(x) + \beta \mathcal{N}(\mu_2, \sigma_2)(x), \tag{3.30}$$

where α and β are the a *priori* probabilities of the two distributions. These assumptions are supported empirically by performing tests on approximately 30 images from various indoor scenes and estimating the cost distributions via forming the histograms, $h[k]$, of the costs, as,

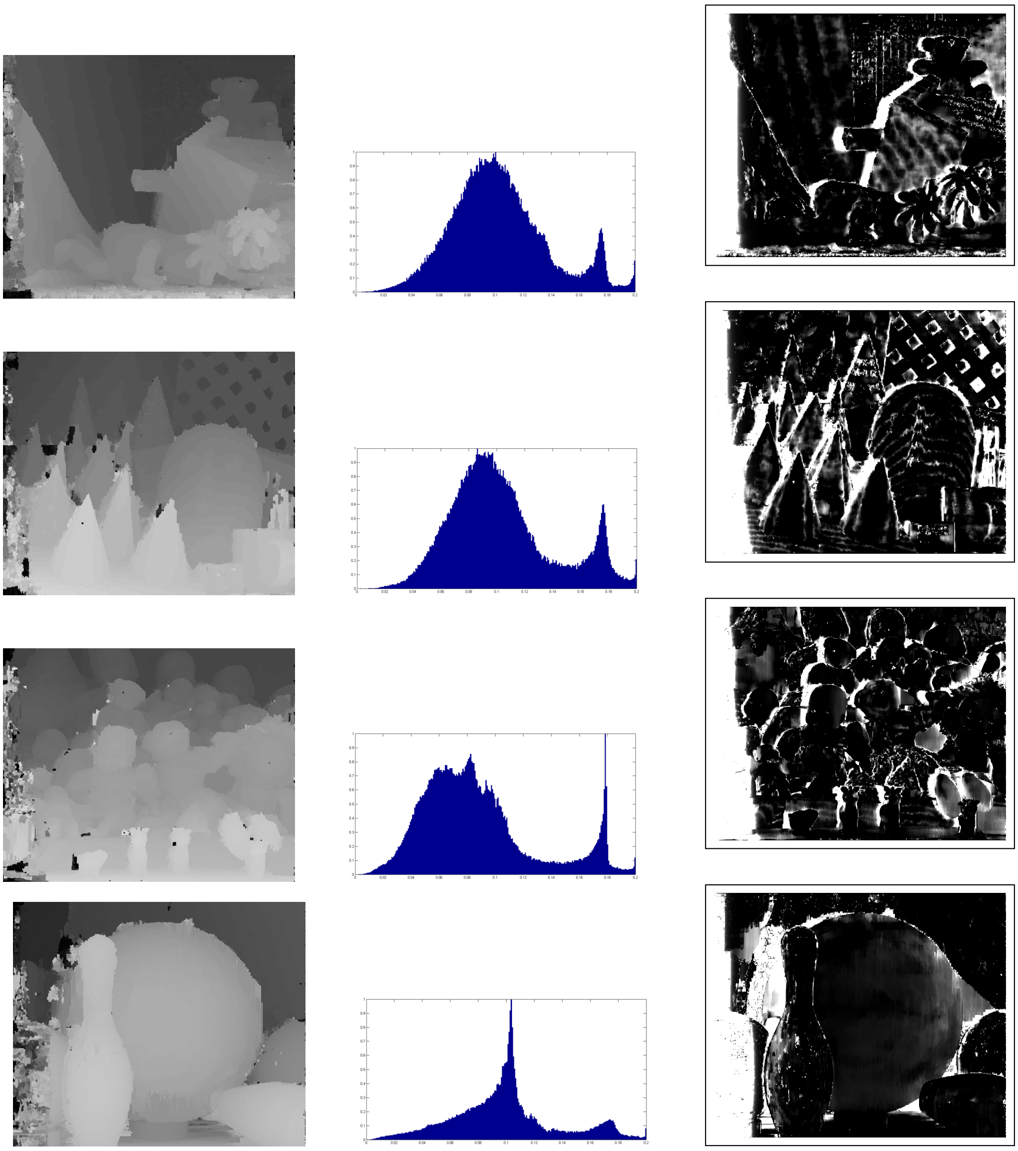
$$\begin{aligned} h[k] &= \frac{1}{\parallel \Omega_{C^{min}} \parallel} \sum_{\forall \bar{p} \in \Omega_{C^{min}}} 1_{\Omega_k}(C^{min}[\bar{p}]),\ 0 \leqslant k \leqslant 255, k \in \mathbb{Z} \\ \Omega_k &= \left[k\,\frac{C^{min}_{max}}{256}, (k+1)\,\frac{C^{min}_{max}}{256} \right),\ C^{min}_{max} = \max_{\bar{p} \in \Omega_{C^{min}}} C^{min}[\bar{p}]\ , \end{aligned} \tag{3.31}$$

where $\Omega_{C^{min}}$ is the domain of the cost function, $C^{min}[\bar{p}]$, given by Eq. 3.29, $\parallel \Omega_{C^{min}} \parallel$ is the total number of pixels in the domain and $1_\Omega(x)$ is the indicator function which takes binary values according to whether its input, x, belong to the set, Ω, as,

$$1_\Omega(x) = \begin{cases} 1, & x \in \Omega \\ 0, & x \notin \Omega \end{cases} . \tag{3.32}$$

$h[k]$ given by Eq. 3.31 gives the frequency of the occurrence of the costs within a range and it can be considered as an estimate of the density function, $g(x)$, given by Eq. 3.30. Sample cost distributions obtained by Eq. 3.31 are presented in Fig. 3.13.

With these assumptions, the disparity estimates can be efficiently divided into two classes according to whether they are reliable or not by estimating the parameters of the cost distribution, $g(x)$, by using the histogram, $h[k]$, of the costs. After obtaining the distribution of the costs by histogram calculation given by Eq. 3.31, an efficient threshold method, *Minimum Error Thresholding* (METH) [40], for bimodal histograms given by Eq. 3.30 is used to obtain distribution parameters of the two classes. In [65], evaluation of many histogram thresholding techniques is available. Among those, the reason for picking METH is not only its efficiency but also its ability to handle histograms with populations of different sizes, and that is the case for the cost histograms. In METH, the threshold, $T^* \in \mathbb{Z}$, to divide the cost histogram, $h[k]$, given by 3.31 into two classes is determined by minimizing a cost function, $J(T)$, with respect to T,



(a) Estimated Disparities (b) Cost Distribution (c) Confidence Costs

Figure 3.13: Disribution of costs (b) and related unreliable regions (c) corresponding to the disparity estimations (c). Brighter regions corresponds to higher costs, namely, unreliable estimates.

where,

$$\begin{aligned} J(T) = & 1 + 2\,[P_1(T)\,\log\sigma_1(T) + P_2(T)\,\log\sigma_2(T)] \\ & - 2\,[P_1(T)\,\log P_1(T) + P_2(T)\,\log P_2(T)]\ , \end{aligned} \tag{3.33}$$

and

$$T^* = \underset{0\leqslant T\leqslant 255}{\arg\min}\, J(T)\ , \tag{3.34}$$

where

$$\begin{aligned} P_1(T) &= \sum_{k=0}^{T} h[k], \\ P_2(T) &= \sum_{k=T+1}^{255} h[k], \end{aligned} \tag{3.35}$$

and

$$\begin{aligned} \sigma_1(T) &= \frac{\sum\limits_{k=0}^{T} (k - \mu_1(T))^2 \; h[k]}{P_1(T)}, \\ \sigma_2(T) &= \frac{\sum\limits_{k=T+1}^{255} (k - \mu_2(T))^2 \; h[k]}{P_2(T)}, \\ \mu_1(T) &= \frac{\sum\limits_{k=0}^{T} k\, h[k]}{P_1(T)}, \\ \mu_2(T) &= \frac{\sum\limits_{k=T+1}^{255} k\, h[k]\, k}{P_2(T)} \; . \end{aligned} \tag{3.36}$$

Finally, once the threshold, T^*, is specified according to Eq. 3.34, a confidence cost can be assigned to each pixel according to the probability of belonging to the class which represents reliable estimates, as,

$$\begin{aligned} C^{Conf}[\bar{p}] &= \begin{cases} 0, & \forall \bar{p} : \underline{C}^{min}[\bar{p}] < \mu_1(T^*) \\ \frac{(\underline{C}^{min}[\bar{p}] - \mu_1(T^*))^2}{2\sigma_1^2(T^*)}, & otherwise \end{cases}, \\ \underline{C}^{min}[\bar{p}] &= 256 \, \frac{C^{min}[\bar{p}]}{C^{min}_{max}} - 1, \end{aligned} \tag{3.37}$$

where $\mu_1(T^*)$ and $\sigma_1^2(T^*)$ are the mean and standard deviation of the class representing the costs of the reliable estimates, respectively, as given by Eq. 3.36, C^{min} is the cost of a disparity assignment at location $\bar{p}$ as given by Eq. 3.29 and C^{min}_{max} is given by Eq. 3.31 and it is the maximum cost value that C^{min} has. The confidence cost, $C^{Conf}[\bar{p}]$, defined by Eq. 3.37 is proportional to the negative logarithm of the probability that the cost, $C^{min}[\bar{p}]$, of the disparity estimate of the pixel $\bar{p}$ comes from the cost distribution of the reliable estimates. Examples of confidence costs assigned to disparity maps are provided in Fig. 3.13c. It can be observed that occluded regions are covered. The obtained confidence cost map is utilized in the following section during transfer of reliable disparities to the finer scale.

3.6.2 Disparity Transfer

The aim of this step is the transfer of reliable disparity estimates to the finer scale as well as handling occlusions. Therefore, window based local approaches introduced in the literature [36, 37, 66] are not appropriate choices for such disparity transfer due to window size dependent performance. In those approaches, the windows are kept small in order not to increase computation load; hence, only a small neighbourhood of the pixels is utilized during the disparity transfer. Yet, small windows might not contain reliable disparity estimates. Especially for occluded regions, the reliable disparity estimates might be present in distant pixels. This situation is illustrated in Fig. 3.12. The occluded regions cannot be handled due to small window size. Increasing the window size can be a solution; yet, with an efficiency drop.

To overcome this problem, the scope of this study is extended from considering only stereo matching methods to considering depth super resolution (DSR) approaches. Two DSR approaches are modified to be utilized in disparity transfer step. Finally, an optimization based method is proposed to deal with the problems DSR based transfer faces with.

3.6.2.1 Depth Super Resolution Approaches

DSR approaches are used to increase the resolution of an available low resolution depth image generally obtained by a depth sensor. Among those, the ones utilizing a high resolution (HR) color image registered with the low resolution depth image [47, 48] can be applied to disparity transfer problem considered in this study; since, the goal is the transfer of the disparity estimates from a coarse scale to a finer scale and a registered finer scale image is available. DSR approaches using registered HR image can be classified into two as global [48] and local [47] methods. The former class is not considered in this study due to its computational complexity. In local methods, recently proposed *Joint Geodesic Upsampling* (JGU) [47] method is proved to be the most efficient method in terms of speed and accuracy. This method performs approximate joint geodesic support filtering in linear time with image size. Therefore, this method is considered to be applicable to disparity transfer problem in this study with some modifications.

Prior to explanation of the JGU based disparity transfer method, the general framework of the DSR methods performing joint filtering should be introduced. Given a low resolution depth image, $D_{\downarrow}$, which is generally obtained from a depth sensor, and the registered HR image, I, a pixel location in the low resolution depth image, $\bar{p}_{\downarrow}$, maps to a unique pixel location in the high resolution image, $\bar{p} = r\,\bar{p}_{\downarrow}$, where r is the scale. The HR depth image, D, of the same resolution with registered image is obtained by performing joint filtering as follows,

$$\begin{aligned} D[\bar{p}] &= \sum_{\bar{q}_{\downarrow}:\bar{q}\in\mathcal{N}_{\bar{p}}} w_{\bar{p},\bar{q}}\, D_{\downarrow}[\bar{q}_{\downarrow}] \\ &\sum_{\bar{q}_{\downarrow}:\bar{q}\in\mathcal{N}_{\bar{p}}} w_{\bar{p},\bar{q}} = 1 \ , \end{aligned} \tag{3.38}$$

where $\mathcal{N}_{\bar{p}}$ is a local neighbourhood of $\bar{p}$ and $w_{\bar{p},\bar{q}}$ is the color adaptive weight as present in EAF. In JGU method, the color adaptive weights are inversely proportional to geodesic distances, $GD[\bar{p},\bar{q}]$, between pixels. In that work, a graph, $\mathcal{G}(e,v)$, consisting of high resolution image grid is formed. The vertices, v, correspond to image pixels, $\bar{p}$ and the edges, e, are inserted between vertices which are neighbours. Neighbouring is considered according to 8-neighbourhood in the image lattice, where neighbourhood is considered as being adjacent from left, right, top, bottom or diagonals. The edges are weighted according to color similarities of the neighbouring pixels in the image, where large weights are introduced for dissimilar pixels. Then the geodesic distances, $GD[\bar{p},\bar{q}]$, are obtained according to Eq. 2.13. In JGU method, the vertices corresponding to mapping of the pixels in the low resolution depth image are considered as seeds and the geodesic distances to the seeds are computed for each pixel; since, only those distances are required according to Eq. 3.38. Using the reference finer scale color image, the geodesic distances are computed with a linear computation time efficiently as explained in [47].

Disparity transfer problem is converted into DSR problem as generating a fine scale disparity image, D, using the disparity image from coarser scale, $D_{\downarrow}$, and the reference image at the fine scale, I, via JGU method. The reliability soft constraint is imposed by assigning initial costs, $VC[\bar{p}]$, which is proportional to the confidence costs, C^{Conf}, given by Eq. 3.37, to the vertices corresponding to seed pixels so that the new geodesic distance, $GD^{new}[\bar{p},\bar{q}]$, becomes,

$$GD^{new}[\bar{p},\bar{q}] = VC[\bar{p}] + GD[\bar{p},\bar{q}],\ \bar{p} : \bar{p} = 2\,\bar{p}_{\downarrow},\ \bar{p}_{\downarrow} \in \Omega_{D_{\downarrow}}\ , \tag{3.39}$$

where $\Omega_{D_{\downarrow}}$ is the domain of the coarse scale disparity image and $VC[\bar{p}]$ is

$$VC[\bar{p}] = \begin{cases} \propto C^{Conf}[\bar{p}_{\downarrow}], & \bar{p} : \bar{p} = 2\,\bar{p}_{\downarrow},\ \bar{p}_{\downarrow} \in \Omega_{D_{\downarrow}} \\ \infty, & otherwise \end{cases}\ . \tag{3.40}$$

In this way, the geodesic distances are affected by the reliability of the disparity estimates.

Apart from the DSR approach, the occlusion filling approach proposed in [16] is also considered as a solution to disparity transfer problem. In that method, occlusions are filled via recursive edge aware filtering according to a confidence map where the occluded regions have zero confidence. This method is also quite efficient and actually similar to JGU method. The disparity transfer problem is handled by this occlusion filling method by defining the required fine resolution confidence map, $Conf$, from coarser resolution confidence cost map ,C^{Conf}, given by Eq. 3.37. The confidences of the pixels corresponding to mapping of the pixels in the low resolution disparity image are set to the values inversely proportional to confidence costs whereas the confidences of the other pixels are set to zero, as,

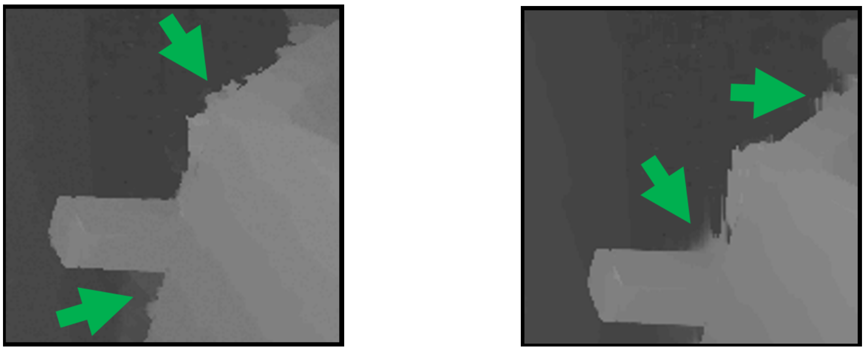
$$Conf[\bar{p}] = \begin{cases} \propto \dfrac{1}{C^{Conf}[\bar{p}_{\downarrow}]}, & \bar{p} : \bar{p} = 2\,\bar{p}_{\downarrow}, \bar{p}_{\downarrow} \in \Omega_{D_{\downarrow}} \\ 0, & otherwise \end{cases} \tag{3.41}$$

Hence, the reliability soft constraint is imposed on the filling process. By performing the occlusion filling explained in [16], a disparity map at the fine scale,

D, is obtained with respect to reliable disparity estimates, as,

$$\begin{aligned} D &= \frac{\mathcal{F}(D^{Conf})}{\mathcal{F}(Conf)} \\ D^{Conf}[\bar{p}] &= \begin{cases} 2\, D_{\downarrow}[\bar{p}_{\downarrow}]\, Conf[\bar{p}], & \bar{p} : \bar{p} = 2\,\bar{p}_{\downarrow},\ \bar{p}_{\downarrow} \in \Omega_{D_{\downarrow}} \\ 0, & otherwise \end{cases} , \end{aligned} \tag{3.42}$$

where $\Omega_{D_{\downarrow}}$ is the domain of the coarse disparity map, $D_{\downarrow}$, $Conf$ is the confidence map given by Eq. 3.41, $\mathcal{F}$ is the filtering process given by Eqs. 2.33 and 2.34 and it is performed by using the fine resolution color image, I, as the guidance image.



(a) Joint Geodesic Upsampling

(b) Recursive EAF based upsampling

Figure 3.14: Illustration of bleeding artefacts near object boundaries.

In Fig. 3.12d and 3.12c, sample disparity estimation results of using the aforementioned approaches for disparity transfer are provided. The improvements over traditional approaches can be observed. Though, those filtering based approaches perform quite well in general in terms of occlusion handling and reliable disparity transfer, as shown in Fig. 3.14, they have *bleeding* problem at the discontinuities where the color transition between foreground and background is not distinctive. At these regions bleeding occurs due to support of the foreground pixels on the background pixels; since, the filtering approaches only consider the influence among color-wise similar pixels and as long as there is path with moderate cost, the pixels support each other. To deal with this problem, a novel method considering such consistencies is proposed in the following section.

3.6.2.2 Proposed Disparity Transfer

In this section, disparity transfer problem is converted into an optimization problem to overcome the issues related with erroneous estimates and to improve the filtering based approaches described in Sec. 3.6.2.1. The objective of the optimization problem is to transfer only the reliable disparity estimates from coarser scale while maintaining the consistency of disparity assignments among pixels within a local region. The former objective serves for error correction and discontinuity preservation; the latter serves to ensure the conditions required for RIP Filter to work; these conditions are explained in Sec. 3.5.

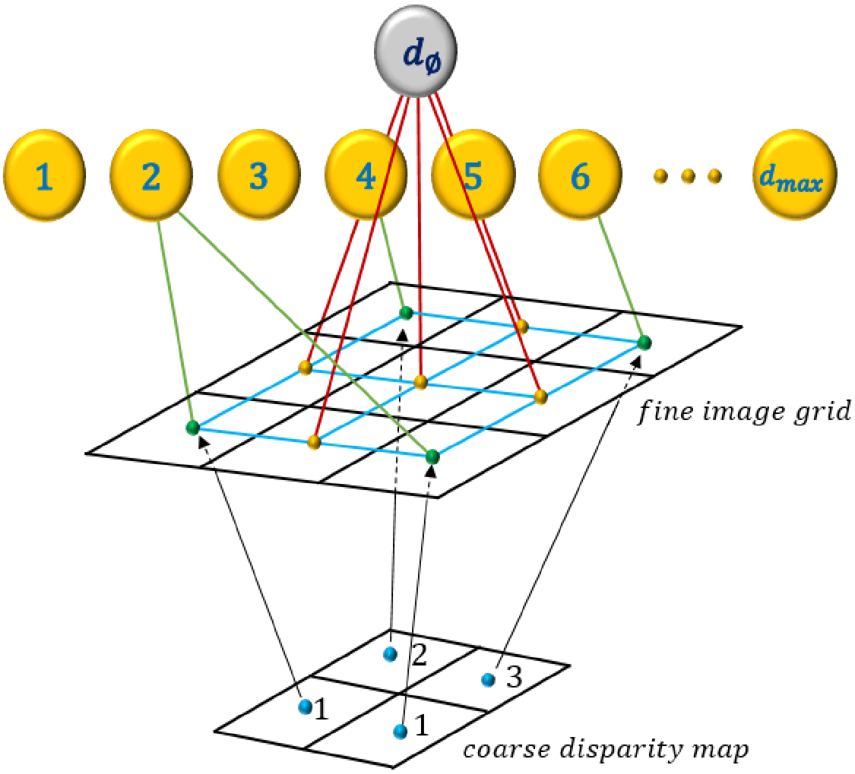


Figure 3.15: Graph formation. Graph consists of the fine image grid and a node for each disparity. A pixel at the fine scale is connected to a disparity if it has a direct correspondence in the coarse scale as given by Eq. 3.43, namely, if it is the sample taken from the fine image as constructing the coarse image. Other pixels are connected to the node, $d_\emptyset$, indicating unknown disparity value.

The proposed method can be simply viewed as the *nearest geodesic neighbour upsampling*. The disparity information of the nearest neighbour is used to form disparity candidate set; yet, there exist some constraints to be satisfied. Therefore, the overall problem can be considered as *finding the geodesic path to the nearest neighbour while satisfying certain constraints.* The problem formulation begins with constructing a graph from the reference image grid at finer scale and the disparity domain. Each node coming from the image grid corresponds to a pixel in the image. Therefore, for the rest of this section, the graph nodes coming from the image grid are denoted as the coordinates of the corresponding pixels, $\bar{p} = (x, y)$, for the sake of clarity. Similarly, each node coming from the disparity domain corresponds to a disparity value and is denoted as simply d. Consequently, there is one node for each pixel in the image at finer scale and one node for each disparity in the disparity search space. Additionally, one more node to represent *unknown disparity*, $d_\emptyset$, is added to the graph to formulate the problem better.

The graph is constructed by forming edges between the nodes. For the nodes, $(\bar{p}, \bar{q})$, coming from the image, an edge, $\xi_{\bar{p},\bar{q}}$, is inserted between two nodes, if they are immediate neighbours. Neighbourhood is considered as being adjacent from left, right, top or bottom (4-connected neighbourhood, $\mathcal{N}^4_{\bar{p}}$). An edge, $\xi_{\bar{p},d}$, between a disparity node, d and a pixel node, $\bar{p}$ is put if there is a corresponding pixel, $\bar{p}_\downarrow$, in the coarser disparity map, $D_\downarrow$, such that

$$\bar{p} = 2\,\bar{p}_\downarrow \; , \tag{3.43}$$

and

$$D_\downarrow[\bar{p}_\downarrow] = \frac{d}{2} \; . \tag{3.44}$$

Otherwise, pixel nodes are connected to unknown disparity node which is denoted as $d_\emptyset$. That is to say, the edge set, $\mathscr{E}_{\bar{p}}$, of a pixel node $\bar{p}$ is:

$$\begin{aligned} \mathscr{E}_{\bar{p}} &= \{\xi_{\bar{p},\bar{q}} : \bar{q} \in \mathcal{N}^4_{\bar{p}}\} \cup \mathscr{E}^{\mathscr{D}}_{\bar{p}} \\ \mathscr{E}^{\mathscr{D}}_{\bar{p}} &= \begin{cases} \xi_{\bar{p},d}, & \exists \bar{p}_\downarrow \in \Omega_{D_\downarrow} : \bar{p} = 2\,\bar{p}_\downarrow, D_\downarrow[\bar{p}_\downarrow] = \dfrac{d}{2} \\ d_\emptyset, & ow \end{cases} \quad , \end{aligned} \tag{3.45}$$

where $\Omega_{D_\downarrow}$ is the image domain of the disparity map at the coarse scale. From Eq. 3.45, it can be observed that a disparity node is connected to a pixel node if the corresponding disparity estimation is directly available from the coarse scale.

Otherwise a pixel node is connected to the node indicating unknown disparity, $d_\emptyset$. An illustrative graph is shown in 3.15.

Following the construction, the graph is converted to a weighted graph by assigning weights to the edges. The edge weight, $w_{\bar{p},\bar{q}}$, between two pixel nodes, $(\bar{p},\bar{q})$, is proportional to color intensity difference, as,

$$w_{\bar{p},\bar{q}} = \frac{1}{2}\left(\frac{max\left\{|I^R[\bar{p}] - I^R[\bar{q}]|, |I^G[\bar{p}] - I^G[\bar{q}]|, |I^B[\bar{p}] - I^B[\bar{q}]|\right\}}{\sigma}\right)^2 , \quad (3.46)$$

where I^R, I^G, I^B are color channels of the reference image and σ is the similarity control parameter obtained by Eq. 3.28. The edge weight, $w_{\bar{p},d}$, between a pixel node and a disparity node, $(\bar{p},d)$, is proportional to the confidence cost, C^{Conf}, which is determined from the costs of the disparity estimates at the coarse scale according to Eq. 3.37, as,

$$w_{\bar{p},d} = C^{Conf}[\bar{p}_\downarrow], \bar{p} = 2\,\bar{p}_\downarrow \ . \quad (3.47)$$

Finally, the edge weight, $w_{\bar{p},d_\emptyset}$, between a pixel node, $\bar{p}$, and the node indicating unknown disparity, $d_\emptyset$, is set to a very large value, as,

$$w_{\bar{p},d_\emptyset} = w_{MAX} \ , \quad (3.48)$$

where w_{MAX} indicates the maximum numeric limit.

Now, the disparity transfer problem can be defined as assigning a disparity to each pixel in the fine resolution image domain, Ω, so that the cost function,

$$E(D) = \sum_{\bar{p}\in\Omega}\left[\underbrace{C[\bar{p}, D[\bar{p}]]}_{\text{data term}} + \overbrace{\sum_{\bar{q}\in N_{\bar{p}}^4} \rho_{\bar{p},\bar{q}}(|D[\bar{p}] - D[\bar{q}]|)}^{\text{smoothness term}}\right] \quad (3.49)$$

gets its minimum. In Eq. 3.49 above, D is the disparity map indicating the disparity assignments of pixels and $\rho_{\bar{p},\bar{q}}$ is the color adaptive penalty term according to disparity assignments of two neighbouring pixels. The formulation of the problem is similar to global stereo matching methods introduce in Sec. 2.1. The data term makes the difference. It is defined to solve transfer of the reliable disparity estimates from coarse scale. For this purpose, it is defined as the geodesic distance of the pixel node $\bar{p}$ to the disparity node $D[\bar{p}]$, as,

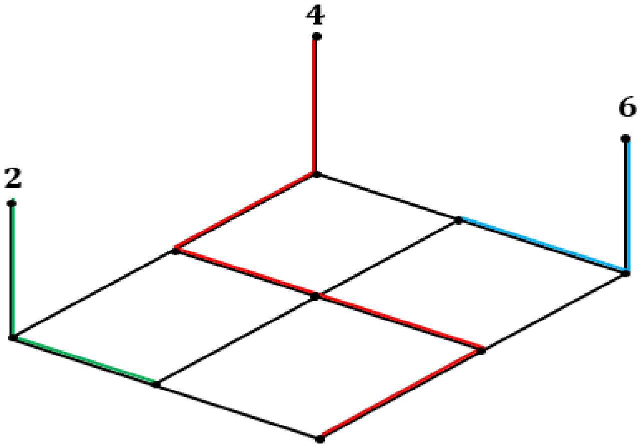
$$C[\bar{p}, D[\bar{p}]] = \textit{geodesic distance between } \bar{p} \textit{ and } D[\bar{p}] \ , \quad (3.50)$$

where geodesic distance is defined in Sec. 2.3 with Eq. 2.13.

With this formulation, the minimization of the cost function defined in 3.49 introduces intensive computational load. It requires preprocessing to construct data term and the computational complexity of this preprocessing is $\mathcal{O}(N \log N)$, where N is the number of pixel nodes. Moreover, the minimization of these kind of energy functions over every possible disparity assignment contradicts the overall goal of the proposed method, and that goal is processing only limited number of disparities for each pixel.

Instead of solving the above optimization problem for disparity transfer, the problem is formulated in an alternative way and solved efficiently under some assumptions. It is assumed that if a disparity value is assigned to a pixel, then all the pixels on the geodesic path between them should also be assigned to that disparity value. This assumption is consistent with the assumption that color-wise similar pixels share same disparity value and enables search space reduction for assignments. Moreover, it is assumed that the geodesic paths between disparity nodes and pixels nodes are in general can be approximated as horizontal and vertical paths. This assumption follows the observation that the estimates at coarse scale are reliable in general and transferring disparity information within a local region is generally sufficient. With this assumption the optimization problem can be solved efficiently by using dynamic programming.

For the alternative problem, a label image, L, is defined for the graph. A label for a pixel at $\bar{p}$ indicates from which neighbour the pixel gets the disparity estimate on the graph, this *label* idea is similar to *back pointers* utilized in graph shortest path problems. Labels can be $\{0, 1, 2, 3, 4\}$, indicating self, left, top, right and bottom respectively. $L[\bar{p}] = 0$ means the pixel at $\bar{p}$ is assigned by the disparity that node $\bar{p}$ is connected to and for the other values the pixel gets the disparity information of the corresponding neighbour. In Fig. 3.16, disparity assignments according to a label map is illustrated. For the bottom right pixel, the label assignment is 2, and this indicates that the bottom right pixel gets the disparity estimate of its upper neighbour. The label of that neighbour is 1 meaning that the disparity estimate of the left neighbour is get. When the labels are traced in



L

0	3	0
2	1	1
0	1	2

D

4	6	6
4	4	4
2	2	4

Figure 3.16: Label image and corresponding disparity assignments

a similar manner, the top left pixel is reached. The label of that pixel indicates that the disparity estimate is 4. Hence, the disparity assignment of the bottom right pixel is 4. In this way, disparity transfer problem can be converted into a labelling problem of 5 possible labels. Minimizing the cost function

$$E_{new}(L) = \sum_{\bar{p}\in\Omega} \left[C[\bar{p}, L[\bar{p}]] + \sum_{\bar{q}\in N^4_{\bar{p}}} \rho_{\bar{p},\bar{q}}(|D^L[\bar{p}, L[\bar{p}]] - D^L[\bar{q}, L[\bar{q}]]|) \right] \tag{3.51}$$

for L, gives the desired disparity transfer to the finer scale. In Eq. 3.51 above, $D^L[\bar{p}, L[\bar{p}]]$ is the resultant disparity assignment to the pixel $\bar{p}$ when the disparity estimate is obtained from the neighbour indicated by $L[\bar{p}]$ and $C[\bar{p}, L[\bar{p}]]$ is the path cost until reaching the disparity node by following label $L[\bar{p}]$. With this convention, the data term and the disparity assignments are obtained via the recursion, as,

$$\begin{gathered} C[\bar{p}, L[\bar{p}]] = \begin{cases} w_{\bar{p},d}, & L[\bar{p}] = 0 \\ w_{\bar{p},\bar{q}} + C[\bar{q}, L[\bar{q}]], & ow \end{cases} \\ \bar{q} = \bar{p} - \bar{r}_{L[\bar{p}]} \ , \end{gathered} \tag{3.52}$$

$$\begin{gathered} D^L[\bar{p}, L[\bar{p}]] = \begin{cases} 2 \cdot D_{\downarrow}[\bar{p}_{\downarrow}], & L[\bar{p}] = 0 \\ D^L[\bar{q}, L[\bar{q}]], & ow \end{cases} \\ \bar{p}_{\downarrow} = \frac{1}{2}\bar{p}, \ \bar{q} = \bar{p} - \bar{r}_{L[\bar{p}]} \ , \end{gathered} \tag{3.53}$$

where $\bar{r}_{L[\bar{p}]}$ is the 2-D vector to move to the coordinates of the immediate neighbour indicated by the label. That is, if $\bar{p} = [x\ y]^T$, then

$$\begin{aligned} \bar{r}_1 &= [1\ 0]^T \\ \bar{r}_2 &= [0\ 1]^T \\ \bar{r}_3 &= [-1\ 0]^T \\ \bar{r}_4 &= [0\ \ -1]^T\ . \end{aligned} \tag{3.54}$$

The last term to be explained is the smoothness term. The purpose of this term is to enforce color-wise similar connected pixels to have similar disparities. Therefore, this terms introduces a penalty cost adaptive to color similarities between pixels in the 4-connected neighbourhood. If two pixels of similar color have different disparities, then a penalty cost is introduced; however, if two pixels are not similar, then the penalty cost is decreased according to the dissimilarity. Explicitly, the penalty function $\rho_{\bar{p},\bar{q}}(\triangle d)$ in Eq. 3.51 is

$$\rho_{\bar{p},\bar{q}}(\triangle d) = \begin{cases} P_1(\triangle d), & \triangle d \leqslant \dfrac{D_C}{2} \\ P_2\,\mathrm{e}^{-w_{\bar{p},\bar{q}}}, & \mathit{otherwise} \end{cases} , \tag{3.55}$$

where D_C denotes the number of disparities in a candidate set. For the disparity differences below the half of the number of the disparity candidates, a linear cost, P_1, increasing with disparity difference is introduced as shown in Fig. 3.17. The purpose is the adaptation to slanted surfaces. For larger disparity differences, a constant cost, P_2, is introduced; however, it is adapted to color-wise similarity of the pixels. The edge weight, $w_{\bar{p},\bar{q}}$, is utilized; since, edge weights obtained by Eq. 3.46 have the pixel similarity information. It is ensured that $P_2\,\mathrm{e}^{-w_{\bar{p},\bar{q}}} \geqslant P_1(\dfrac{D_C}{2})$.

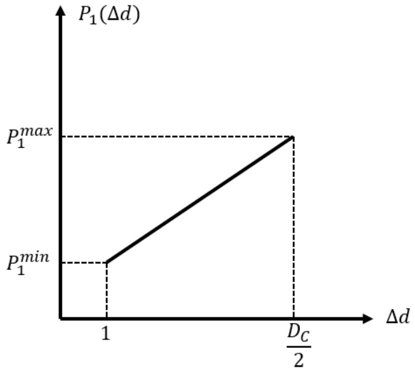


Figure 3.17: Linear penalty term

Finally, with the minimization of the function in Eq. 3.51, disparity estimates of the reliable pixels are propagated through the the image and the color-wise similar pixels within a neighbourhood are assigned to similar disparities. While colour based edge weights provide edge-aware propagation, color adaptive penalty term makes sure that color-wise similar pixels have similar disparity assignments.

The minimization is performed approximately by using dynamic programming for the sake of efficiency. The graph is processed in horizontal and vertical directions, sequentially. The cost function given in Eq. 3.51 is defined for each row and column of the graph grid composed of node pixels. In this way, the number of possible labels, which are the labels indicating the self assignment and the assignment of preceding neighbour corresponding to the pass direction of the dynamic programming, are reduced to two. Therefore, the the label set can be simply considered as $\{0, 1\}$, where 0 means the disparity estimate of the node itself is assigned to the current node and 1 means disparity estimate of the preceding node is assigned according to the progression direction.

Using dynamic programming, the minimization according to a direction is performed as,

$$\begin{aligned}
&e[\bar{p}, 0] = \\
&\quad w_{\bar{p}, D^L[\bar{p},0]} + min\{e[\bar{p}-\bar{r}, 0] + \rho_{\bar{p},\bar{p}-\bar{r}}(|D^L[\bar{p},0] - D^L[\bar{p}-\bar{r},0]|), \\
&\qquad\qquad\qquad\qquad e[\bar{p}-\bar{r}, 1] + \rho_{\bar{p},\bar{p}-\bar{r}}(|D^L[\bar{p},0] - D^L[\bar{p}-\bar{r},1]|)\} \\
&e[\bar{p}, 1] = \\
&\quad w_{\bar{p},\bar{p}-\bar{r}} + min\{e[\bar{p}-\bar{r}, 0] + C[\bar{p}-\bar{r}, 0], \\
&\qquad\qquad\qquad e[\bar{p}-\bar{r}, 1] + C[\bar{p}-\bar{r}, 1]\} \ ,
\end{aligned} \tag{3.56}$$

where the edge weights, $w_{\bar{p},\bar{q}}$, $w_{\bar{p},d}$, and the penalty term, $\rho_{\bar{p},\bar{q}}$, are given by Eqs. 3.46, 3.47 and 3.55, respectively and $\bar{p} - \bar{r}$ is the preceding node according to progression direction. Cost of each individual node, e, contains cost knowledge of the preceding nodes. Thanks to recursive relations given in Eqs. 3.52 and

3.53, the values of $D^L[\bar{p},1]$ and $C[\bar{p},1]$ can be computed as,

$$\begin{aligned} C[\bar{p},1] &= w_{\bar{p},\bar{p}-\bar{r}} + C[\bar{p}-\bar{r},i] \\ D^L[\bar{p},1] &= D^L[\bar{p}-\bar{r},i] \\ \text{where } i &= \underset{i\in\{0,1\}}{\arg\min}\{C[\bar{p}-\bar{r},i] + e[\bar{p}-\bar{r},i]\} \ . \end{aligned} \tag{3.57}$$

After computing costs, e, for every node according to Eq. 3.56, the disparities, D, and the corresponding geodesic distances, C^{GD}, are assigned starting from the end node as,

$$\begin{aligned} D[\bar{p}] &= D^L[\bar{p},i] \\ C^{GD}[\bar{p}] &= C[\bar{p},i] \\ \text{where } i &= \underset{i\in\{0,1\}}{\arg\min}\, e[\bar{p},i] \ , \end{aligned} \tag{3.58}$$

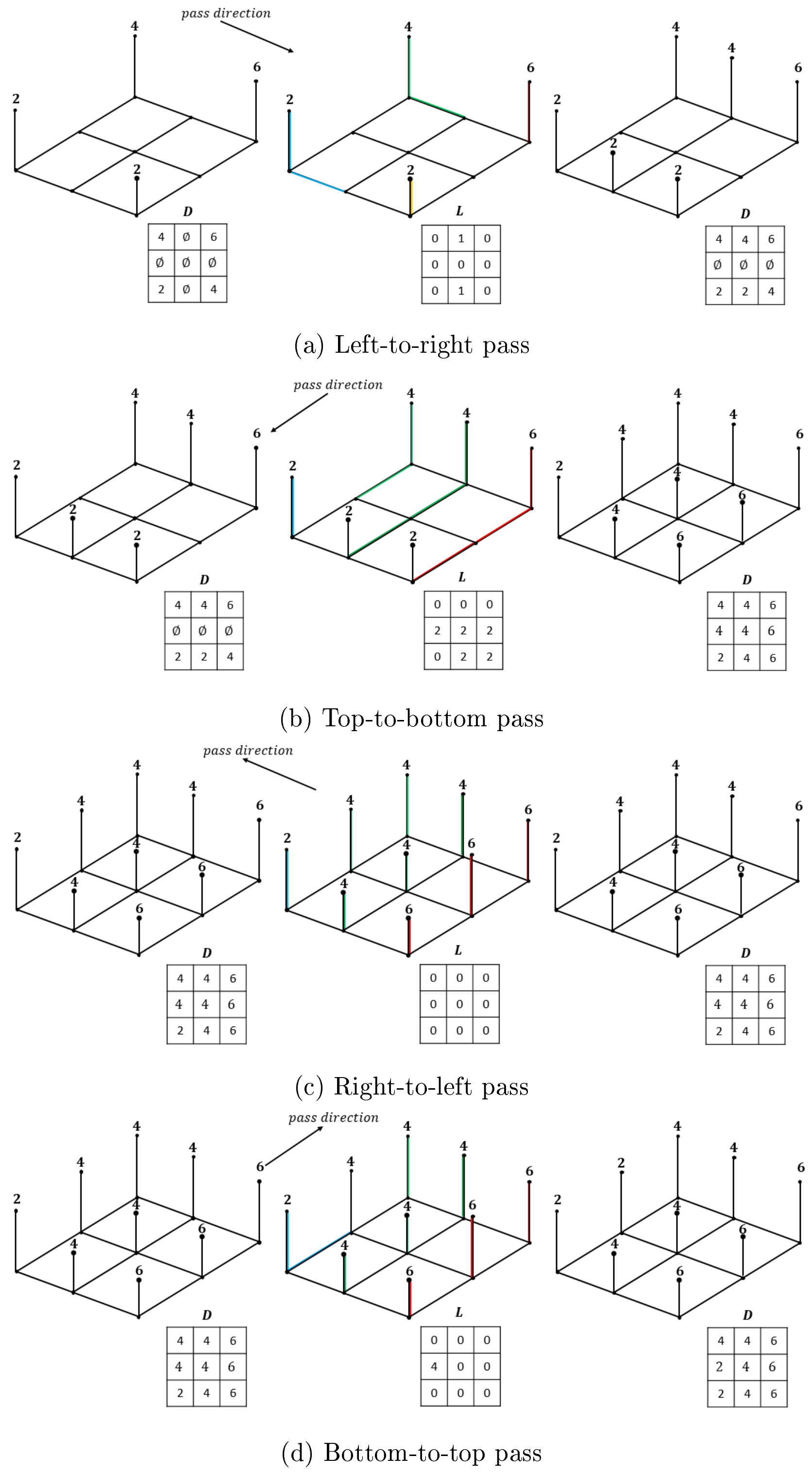
and after each assignment, the knowledge is back propagated by updating the cost of the preceding node, $e[\bar{p}-\bar{r},0]$, as,

$$\begin{aligned} e^{updated}[\bar{p}-\bar{r},0] &= e[\bar{p}-\bar{r},0] + \rho_{\bar{p}-\bar{r},\bar{p}}(|D^L[\bar{p}-\bar{r},0] - D[\bar{p}]|) \\ e^{updated}[\bar{p}-\bar{r},1] &= e[\bar{p}-\bar{r},1] \ . \end{aligned} \tag{3.59}$$

Finally, the disparity assignment is performed according to Eq. 3.58 by using $e^{updated}$ instead of e. It should be noted that while computing $C[\bar{p},1]$ and $D^L[\bar{p},1]$ in Eq. 3.57, it is assumed that the information coming from proceeding node does not effect the result, or even if it does, ignoring this effect does not change the overall solution, considerably.

To approximately solve the optimization problem given in Eq. 3.53, a dynamic programming approach presented in Eq. 3.56 is applied by passing the graph in horizontal and vertical directions. Between each pass, the graph is updated according to disparity map result, $D^{updated}$, of the last pass. That is to say, every edge between pixel nodes and disparity nodes in the previous graph are removed. Then, an edge between a pixel node and a disparity node is inserted according to the result of the dynamic programming. Hence, the edge set, $\mathscr{E}_{\bar{p}}$, of a pixel, $\bar{p}$, becomes

$$\mathscr{E}_{\bar{p}} = \{\xi_{\bar{p},\bar{q}} : \bar{q} \in \mathcal{N}^4_{\bar{p}}\} \cup \{\xi_{\bar{p},d} : d = D^{updated}[\bar{p}]\} \ , \tag{3.60}$$



D

4	Ø	6
Ø	Ø	Ø
2	Ø	4

L

0	1	0
0	0	0
0	1	0

D

4	4	6
Ø	Ø	Ø
2	2	4

(a) Left-to-right pass

D

4	4	6
Ø	Ø	Ø
2	2	4

L

0	0	0
2	2	2
0	2	2

D

4	4	6
4	4	6
2	4	6

(b) Top-to-bottom pass

D

4	4	6
4	4	6
2	4	6

L

0	0	0
0	0	0
0	0	0

D

4	4	6
4	4	6
2	4	6

(c) Right-to-left pass

D

4	4	6
4	4	6
2	4	6

L

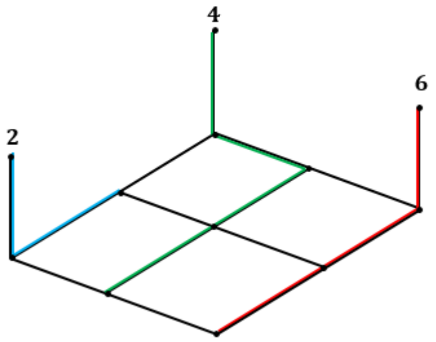
0	0	0
4	0	0
0	0	0

D

4	4	6
2	4	6
2	4	6

(d) Bottom-to-top pass

Figure 3.18: Graph update example. The coloured paths in the middle column indicate the estimated minimum cost paths. After each pass, the graphs are updated according to the minimum cost paths indicated by label image L.



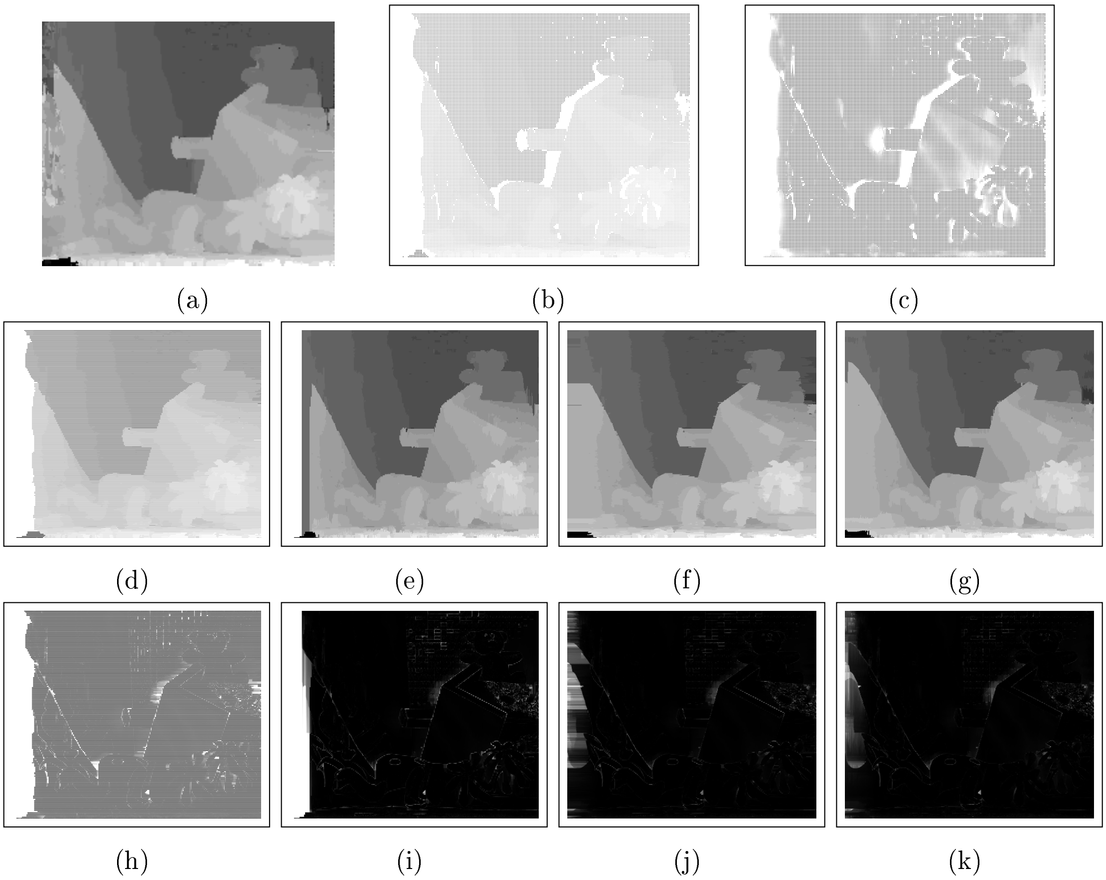
L

0	1	0
4	2	2
0	2	2

D

4	4	6
2	4	6
2	4	6

Figure 3.19: Overall solution corresponding to initial graph given in Fig. 3.18a.


(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Figure 3.20: Propagation of the disparity estimates from coarse scale to fine scale by the proposed method. (a) Coarse disparity map. (b) Initial disparity assignments, brightest regions are unknown regions. Occluded regions are marked as unknown. (c) Confidence costs of the initial assignments. (d-e-f-g) disparity assignments after each pass; left-to-right, top-to-bottom, right-to-left and bottom-top-top respectively. (h-i-j-k) Confidence costs of the assignments after each pass, respectively.

where $D^{updated}$ is the disparity map result of the last pass and it is obtained by Eq. 3.58. The edge weights, $w_{\bar{p},d}$, are set according to the computed costs in Eq. 3.58, as,

$$w_{\bar{p},d} = C^{GD}[\bar{p}] \ . \tag{3.61}$$

The passes are performed sequentially as left-to-right, top-to-bottom, right-to-left and bottom-to-top. In Fig. 3.18, a sample optimization procedure is presented with graph updates. After the left-to-right pass, the obtained disparity map is utilized to update the graph as shown in Fig. 3.18a. Then, the top-to-bottom pass is performed on this updated graph. This process is continued until all four passes are performed. In this way, while approximating the geodesic paths, the disparity estimates which comes from each neighbour are considered for a pixel node in the optimization process and reliable disparity estimates are propagated through the graph. In Fig. 3.19, the overall result of the minimization process, which consists of 4 passes, of the cost function given by Eq. 3.51 is illustrated. In Fig. 3.20, the propagation of the estimates after each pass is presented. The graphs after each pass and the edge weights to the disparity nodes are provided via gray-scale images.

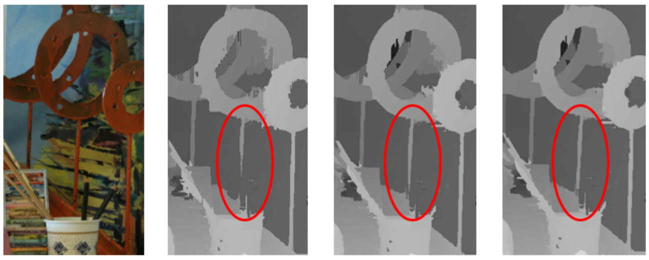


Figure 3.21: Thin structure recovery via multiple iterations on disparity transfer. From left to right: Reference image, disparity map with 1 iteration, disparity map with 2 iterations, disparity map with 3 iterations.

Proposed disparity transfer method can be applied iteratively so that the reliable disparity estimates from more complex paths can be retrieved. Hence, thin object loss problem can be handled as long as a part of thin object is connected to a reliable estimate. Fig. 3.21 depicts such a situation. As it can be observed thin objects are lost as a consequence of the CTF scheme. Iterative application of the transfer the proposed transfer iteratively can recover these structures via propagating the estimates from the objects that thin objects are connected.

3.6.2.3 Candidate Disparity Generation and Disparity Assignment

Given a disparity map at a coarse scale, $D_{\downarrow}$, the finer scale disparity map, D, is obtained using the methods presented in either Sec. 3.6.2.1 or 3.6.2.2 via Eqs. 3.38, 3.42 or 3.58. Using this disparity map, D, the candidate set of each pixel, $\mathcal{D}_{\bar{p}}$ for a given candidate number, D_C, is formed as,

$$\mathcal{D}_{\bar{p}} = \{d : |D[\bar{p}] - d| \leqslant \frac{D_C}{2}\},\ d \in \mathbb{Z}. \tag{3.62}$$

Once the disparity hypotheses, $\mathcal{D}_{\bar{p}}$, are specified for the pixels in the finer scale, the sparse matching cost volume, C, is obtained according to Eq. 3.7 by using the finer scale left and right stereo image pairs from the left and right image pyramids given by Eq. 3.1. Then the disparities are assigned according to aggregated costs, C^f, which are obtained by 3.15, as,

$$D^{final}[\bar{p}] = \underset{d \in \mathcal{D}_{\bar{p}}}{\arg\min}\, C^f[\bar{p}, d]\ . \tag{3.63}$$

If D^{final} is the disparity assignments at the finest scale, then it is the result of the proposed algorithm. Otherwise, it is used to generate disparity hypotheses, $\mathcal{D}_{\bar{p}}$, for the pixels in the finer scale as given by Eq. 3.62.

3.7 Complexity Analysis

In this section the complexity of the method is analysed and its linear time complexity with image size is theoretically derived. Then, regarding to complexity-accuracy trade-off, a pyramid level is determined and it is shown that without breaking the efficiency the pyramid levels can be reduced from the optimal pyramid level for the fastest computation. Complexity is to be analysed in two parts. First part discusses the complexity of the proposed cost aggregation method and the second part proves that the computational complexity of the proposed disparity estimation method is not affected by the size of the disparity search space.

3.7.1 RIP Filter Complexity

As discussed in Sec. 3.5.2, the prediction function introduces additional complexity. On the other hand, if candidates are generated in a structured way, then this complexity can be reduced. If D_C is the number of disparity candidates for each pixel and N is the size of the image; then cost aggregation is performed in $\mathcal{O}(N \cdot D_C) + \mathcal{O}(N \cdot \log(D_C))$ time complexity. This additive complexity comes from extra comparison operations during nearest neighbour search which are far less complex than operations like multiplication. Moreover, generally a small number of disparity candidates are used such as 5, 7, 11. Therefore, additive effect of $\mathcal{O}(N \cdot \log(D_C))$ can be considered as a constant additive effect regardless of D_C and can be neglected even for small values of D_C. Consequently, the overall complexity of the cost aggregation step is $\mathcal{O}(N \cdot D_C)$.

3.7.2 Coarse to Fine Approach Complexity

The complexity is derived based on an initial disparity search range variable; then it is optimized according to initial search range in order to determine optimal pyramid level. In the analysis, the complexity introduced from pyramid building and disparity estimate transfer is ignored since they introduce an additive complexity with $\mathcal{O}(N \cdot C)$ where N is the image size and C is some constant. This complexity is already independent of the disparity search range, D, and this complexity can be encoded in number of disparity candidates, D_C, as analysing the complexity.

Given D as the size of disparity range at the finest scale and D_0 as the size of disparity range at the top level of the image pyramid, the top level, n, of the pyramid is determined as,

$$n = \lfloor \log_2 \frac{D}{D_0} \rfloor \ . \tag{3.64}$$

For a given number of the disparity candidates at each level, D_C, which is lower than the disparity search range at the top level, D_0; the computation complexity,

C_t is derived as,

$$C_t \propto \sum_{k=0}^{n-1} \left(\frac{N\,D_c}{4^k} \right) + \frac{N\,D_0}{4^n} \ . \tag{3.65}$$

For the sake of simplicity, $\frac{D}{D_0}$ is assumed to be a power of 2. Using the *geometric series formula*, which is

$$\sum_{k=0}^{n-1} a\,r^k = a\,\frac{1-r^n}{1-r} \ , \tag{3.66}$$

and from Eq. 3.64, the first term on the left hand side in Eq. 3.65 becomes

$$\sum_{k=0}^{n-1} \left(\frac{N\,D_c}{4^k} \right) = \frac{4}{3}\,N\,D_c \left[1 - \left(\frac{1}{4} \right)^{\log_2 \frac{D}{D_0}} \right] = \frac{4}{3}\,N\,D_c \left[1 - \left(\frac{D_0}{D} \right)^2 \right] \ . \tag{3.67}$$

Similarly, the second term on the left hand side in Eq. 3.65 becomes

$$\frac{N\,D_0}{4^{\log_2 \frac{D}{D_0}}} = \frac{D_0^3}{D^2} \ . \tag{3.68}$$

Then Eq.3.65 becomes:

$$C_t \propto N \cdot \left[\frac{4}{3} D_C \left[1 - \left(\frac{D_0}{D} \right)^2 \right] + \frac{D_0^3}{D^2} \right] \ , \tag{3.69}$$

$$C_t \propto f(D_0) \ , \ f(D_0) = \frac{1}{D^2} D_0^3 - \frac{4D_C}{3D^2} D_0^2 + \frac{4D_C}{3} \ . \tag{3.70}$$

As $D \to \infty$, complexity, C_t, becomes proportional to $N\,\frac{4D_C}{3}$, which introduces $\mathcal{O}(N)$ complexity independent of the disparity search range, D.

3.7.3 Optimal Pyramid Level

In practice, D is finite. Therefore, D_0 should be selected as $D_0 \ll D$ in order to reduce complexity introduced in Eq. 3.69 to $\mathcal{O}(N)$. However, decreasing D_0 too much results in an increase in the complexity as shown in Fig. 3.22. The optimal value of D_0 can be determined from the function in Eq. 3.70 by seeking the point where $\nabla_{D_0^{opt}} f = 0$:

$$\nabla_{D_0} f = \frac{3}{D^2} D_0^2 - \frac{8D_C}{3D^2} D_0 = 0 \ . \tag{3.71}$$

For a given D and D_C, the function in Eq. 3.70 has a unique global minimum subject to $D_0 > 0$, as,

$$D_0^{opt} = \frac{8}{9} D_c \ , \tag{3.72}$$

$$f(D_0^{opt}) = \frac{4D_C}{3} - \frac{2^8 D_C^3}{3^6 D^2} \ . \tag{3.73}$$

Eq. 3.73 shows that if D_0 is at optimal value, the complexity is $\mathcal{O}(N)$ as long as $D_c \ll D$.

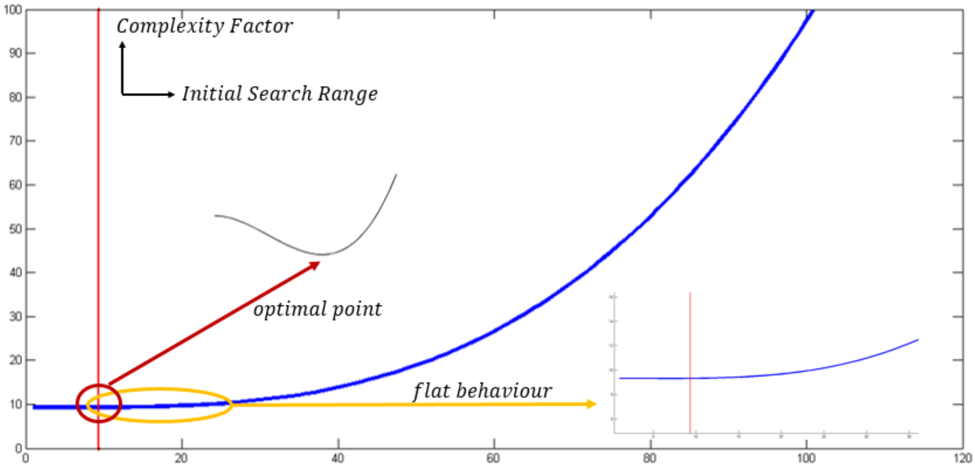


Figure 3.22: Behaviour of the function of the complexity factor due to disparity search range derived in Eq. 3.70 when $D = 100$ and $D_C = 7$.

Picking $D_0 = D_0^{opt}$ results in a large number of levels in the image pyramid for high resolution images. This sometimes causes a loss of fine structures, which cannot be recovered, due to downsampling. If the behaviour of the cost function in Eq. 3.70 is examined, it can be observed that there is a flat behaviour near optimal point until some larger point; and this is illustrated in Fig. 3.23. This

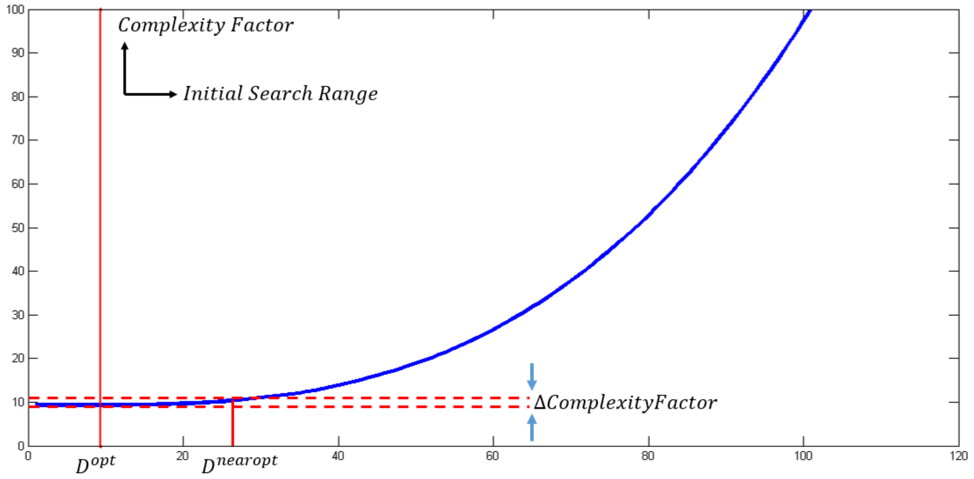


Figure 3.23: Increase in complexity in case of deviation from optimal point.

flat behaviour is related with the coefficients of the cubic function. From this observation, it can be concluded that increasing D_0 beyond optimal value up to some value causes negligible deviation from the optimal complexity. Hence, D_0 can be increased to gain from accuracy without loosing from complexity. How much D_0 should be increased can be determined by solving the following cubic equation for $\triangle D_0$ for a given tolerance, Δf, to increase in complexity as shown in Fig. 3.23:

$$\begin{aligned} \frac{1}{D^2}(D_0^{opt} + \Delta D_0)^3 - \frac{4D_C}{3D^2}(D_0^{opt} + \Delta D_0)^2 + \frac{4D_C}{3} \\ = f(D_0^{opt}) + \Delta f \ , \end{aligned} \tag{3.74}$$

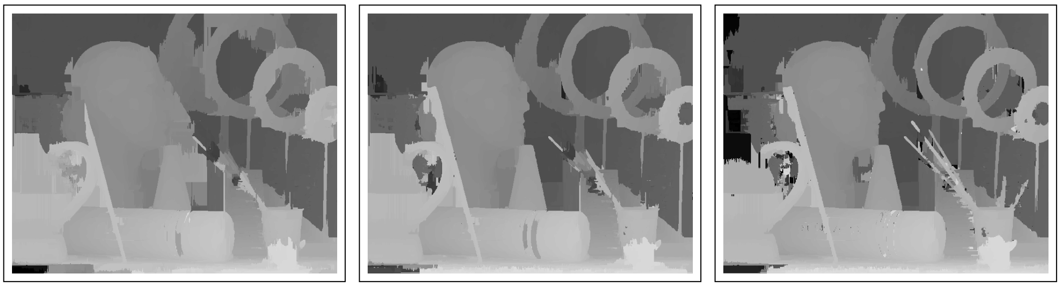
$$D_0^{nearopt} = D_0^{opt} + \Delta D_0 \ , \tag{3.75}$$

where f is the complexity function given by Eq. 3.70. Increasing the initial disparity search range beyond the optimal search range, D_0^{opt}, results in an increase in the complexity factor due to the disparity search range. This increase is denoted as Δf in Eq. 3.74. Once the tolerated increase is specified, near optimal initial search range can be obtained. In Fig. 3.24 comparison of the accuracy and the complexity of the disparity estimation under the two different tolerances is shown. Increasing the tolerance results in increase in the initial disparity search range. Hence the maximum pyramid level is decreased. It can be observed that selecting D_0 as near optimal value given by Eq. 3.75 improves quality of the result with a negligible increase in complexity. The disparities of the thin structures are better estimated by using larger initial disparity search ranges than the optimal disparity search range.

3.8 Experimental Results

The proposed method is tested in several aspects including disparity estimation accuracy, algorithm time complexity and robustness under various stereo scenes. Following the experiments, the performance results of the proposed method are compared with the prior art.

Middlebury test bench [62] enables evaluation and comparison of disparity estimation methods. The evaluations and comparisons are based on four stereo



(a) $\Delta f = 0$: t_{opt} (b) $\Delta f = 2$: $1.01 \cdot t_{opt}$ (c) $\Delta f = 5$: $1.3 \cdot t_{opt}$

Figure 3.24: Results under different initial disparity search ranges that are picked according to the given tolerance to increase in the optimal complexity, Δf. The relative execution times of the disparity estimations with different initial disparity search ranges are presented after colons. t_{opt} is the execution time for optimal initial search range. Optimal search range causes higher pyramid levels and loss of thin structures. Picking near optimal values reduces maximum pyramid level and improves results without degrading the efficiency.

image pairs ground truths of which are available (Fig. 3.26) and only the estimation accuracy is evaluated. This environment is used to compare the disparity estimation accuracy of the proposed method with the other methods in the literature.

Information Permeability [17] and *Domain Transform* [60] based stereo matching methods are implemented for the time complexity comparison purpose. These methods are reported to be the fastest stereo matching methods available in the literature. Proposed method and these methods are implemented by using the same data structures and memory management scheme so that a fair comparison regarding to computation time can be conducted. Using the results of the published studies [16, 21, 36, 37, 49, 50, 54, 56, 57, 60, 81, 87], the relative computational complexities of the existing methods are retrieved and in this way, the computational complexity of the proposed method are compared with the previous works.

Moreover, additional experiments are performed on the extended data set provided by Middlebury [62] in order to examine the performance of the proposed method under various stereo scenes. The results of these experiments give insight into the robustness of the proposed method. Apart from the comparisons according to Middlebury test bench consisting of four stereo pairs, the proposed method is compared with the previous works which have the performance results using the extended data set.

Finally, the proposed method is analysed according to the test results. The cases in which the proposed method is strong and the cases in which the proposed method is weak are discussed. The linear time complexity of the method is validated and complexity-accuracy trade-off is provided according to number of disparity candidates, maximum pyramid level and disparity information transfer method.

3.8.1 Parameter Setting and Visual Results

The internal parameters of the algorithmic blocks are fixed during all of the experiments. *Matching Cost Parameters* according to Eq. 3.6 are set as, $\alpha = 0.4$, $Census\ Window\ Size = 3 \times 3$ and $T = 0.15$. All smoothness control parameters (or decay control parameters) appear in the equations, σ, are automatically determined according to Sec. 3.5.3. The number of disparities in the candidate set (appears as D_C in Eq. 3.62) is fixed to 7 and the initial disparity search range, D_0, is determined according to Eq. 3.74 by setting the tolerance as $\Delta f = 1$. According to D_0, the maximum pyramid level, n, in Eq. 3.1 is obtained by Eq. 3.64. Finally, the parameters of the proposed *disparity transfer* method (referring to Eq. 3.55 and Fig. 3.17) are set as, $P_1^{min} = 0.25$, $P_1^{max} = 1.25$ and $P_2 = 3$. These parameters are obtained empirically following to a grid search for the best parameters on the data set composed of four images in the test bench and fixed for the tests conducted on the extended data set.

The parameters for the geodesic support filtering [47] and recursive edge aware filtering [16] based disparity transfer are set according to parameters that are suggested by the authors of these studies [16, 47].

The inputs to the proposed algorithm are the left, I_L, and the right, I_R, image of the stereo image pair, maximum possible disparity, $d_{max} \in \mathrm{Z}$, and the scale factor, $s \in \mathrm{Z}^+$. Scale factor, s, is used to scale the disparity estimates to visualize the results better, as,

$$D^{visual}[x,y] = s\,D^{final}[x,y],\ 0 \leqslant D^{final}[x,y] \leqslant d_{max},\ D^{final}[x,y] \in \mathrm{Z}\ , \quad (3.76)$$

where D^{final} is the disparity estimation result of the proposed algorithm and it is obtained by Eq. 3.63. An example of a visual disparity estimation result, D^{visual}, is provided in Fig. 3.25. The same coordinate system convention which is presented in Fig. 3.25 is used in all of the visual results which are to be presented in this section by Figs. 3.26, 3.29, 3.32 and 3.34-3.41.

For the results presented in Fig. 3.26, x and y coordinate ranges are $x \in [0,383]$, $y \in [0,287]$; $x \in [0,433]$, $y \in [0,382]$; $x \in [0,359]$, $y \in [0,374]$; $x \in [0,449]$, $y \in [0,374]$; $x, y\ \in \mathrm{Z}$, respectively, from left to right. (d_{max}, s) parameters are $(28,8)$, $(20,8)$, $(53,4)$, $(55,4)$, respectively, from left to right.

For the results presented in Fig. 3.32, x and y coordinate ranges are $x \in [0,1389]$, $y \in [0,1109]$, $x, y\ \in \mathrm{Z}$. (d_{max}, s) parameters are $(255,1)$.

For the results presented in Figs. 3.34-3.41, x and y coordinate ranges are $x \in [0, W-1]$, $y \in [0, H-1]$, $x, y\ \in \mathrm{Z}$, where W and H are the width and height of the input images. The disparity transfer method for those results is the proposed DP based disparity transfer method.

For the results presented in Figs. 3.34, $(W \times H)$ are (620×555), (432×381), (430×381), (433×381), (626×555), respectively from top to bottom. (d_{max}, s) parameters are $(69,2)$, $(17,8)$, $(17,8)$, $(20,8)$, $(86,2)$, respectively, from top to bottom.

For the results presented in Figs. 3.35, $(W \times H)$ are (650×555), (695×555), (656×555), (671×555), (284×216), respectively from top to bottom. (d_{max}, s) parameters are $(114,2)$, $(111,2)$, $(91,2)$, $(116,2)$, $(29,8)$, respectively, from top to bottom.

For the results presented in Figs. 3.36, $(W \times H)$ are (683×555), (695×555), (665×555), (435×383), (638×555), respectively from top to bottom. (d_{max}, s) parameters are $(93,2)$, $(109,2)$, $(80,2)$, $(21,8)$, $(85,2)$, respectively, from top to bottom.

For the results presented in Figs. 3.37, $(W \times H)$ are (638×555), (434×380), (653×555), respectively from top to bottom. (d_{max}, s) parameters are $(84, 2)$, $(18, 8)$, $(109, 2)$, respectively, from top to bottom.

For the results presented in Figs. 3.38, $(W \times H)$ are (620×555), (695×555), (665×555), (650×555), (650×555), respectively from top to bottom. (d_{max}, s) parameters are $(78, 2)$, $(111, 2)$, $(99, 2)$, $(114, 2)$, $(101, 2)$, respectively, from top to bottom.

For the results presented in Figs. 3.39, $(W \times H)$ are (698×555), (635×555), (686×555), respectively from top to bottom. (d_{max}, s) parameters are $(104, 2)$, $(98, 2)$, $(108, 2)$, respectively, from top to bottom.

For the result presented in Figs. 3.40, $(W \times H)$ is (671×555) and (d_{max}, s) parameters are $(101, 2)$.

For the results presented in Figs. 3.41, $(W \times H)$ are (641×555), (695×555), (626×555), (650×555), (650×555), respectively from top to bottom. (d_{max}, s) parameters are $(106, 2)$, $(112, 2)$, $(115, 2)$, $(97, 2)$, $(98, 2)$, respectively, from top to bottom.

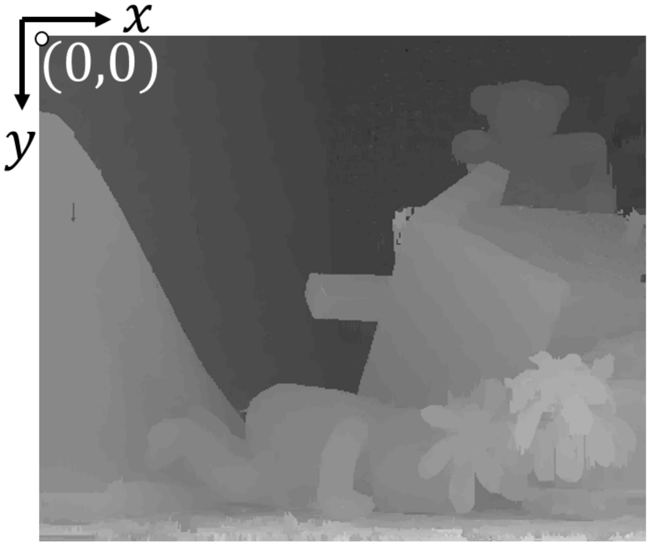


Figure 3.25: Example of a visual disparity estimation result, D^{visual}. Each value, $D^{visual}[x, y]$, $x \in [0, 359]$, $y \in [0, 374]$, $x, y \ \in$ Z, is obtained according to Eq. 3.76 with the parameter setting presented in Sec. 3.8.1. The input left, I_L, and right, I_R, images are *Teddy* [62] stereo pairs and d_{max} and s in Eq. 3.76 are 53 and 4, respectively.

3.8.2 Disparity Estimation Accuracy

In this part, the disparity estimation accuracy of the proposed method is examined as a consequence of the results of the tests. Two tests are conducted on different data sets. First data set is composed of the four stereo image pairs provided by Middlebury test bench [62] and the second data set is the extended version of this data set. The ground truth disparity maps of the stereo image pairs in those data sets are available. The accuracy of the disparity estimation is evaluated according to the average number of bad pixels in the disparity maps. A pixel is regarded as a bad pixel if its disparity estimation error is larger than 1 pixel. The disparity estimation error is determined from the ground truths and it is the absolute difference of the estimate and ground truth value.

For the comparison with the previous work, the evaluation result of the proposed method for the four stereo image pairs is provided together with the results of the existing methods in Table 3.1 and in Table 3.2 for the disparity estimation accuracy considering all regions and non-occluded regions, respectively. The state-of-the-art local methods [17,21,22,34–37,49,50,53–57,60,66,81,82,87,91] are included in the comparison, since the proposed method is a local approach. Moreover, well-known semi-global methods [31,43,76] are also included due to their similar scanline based approaches. The proposed methods are denoted as Proposed(NGN), Proposed(REAF), Proposed(GSF), Proposed(MG) and Proposed(NN) according to the disparity transfer method they utilize. NGN stands for the proposed *nearest geodesic neighbour* approach, REAF and GSF represent *recursive edge aware filtering* and *geodesic support filtering* based information transfer approaches respectively. Finally, MG and NN are the nearest neighbour approaches utilizing matching costs and color-wise similarities, respectively.

According to the evaluation results provided in Table 3.1 for the disparity estimates at all regions, the proposed method lies in the mid-range, however it can be observed that the performances of the methods are pretty close to each other. Except for the top performer and the bottom three methods, the percentage of the erroneous pixels is between 5 and 8. The proposed method has 77% more erroneous estimates relative to the top performer, and this corre-

Table3.1: Disparity estimation accuracy comparisons of the selected methods for all pixels in Middlebury evaluation data set.

Method	**Tsukuba**	**Venus**	**Teddy**	**Cones**	**Avg. %** **Bad Pixels**
PatchMatchFilter [50]	2.04	0.49	5.87	6.8	3.80
CrossLMF [49]	2.78	0.38	10.6	7.82	5.40
HistoAggr2 [56]	2.3	0.46	11.13	7.78	5.42
DomainTfm [60]	2.1	0.45	11.5	7.82	5.47
CostFilter [35]	1.85	0.39	11.8	8.24	5.57
NonLocal [81]	1.85	0.42	11.6	8.45	5.58
GeoDiff [22]	2.35	0.82	11.13	8.33	5.66
SegmentTree [54]	1.68	0.3	11.9	8.82	5.68
P-LinearS [21]	1.67	0.89	12	8.44	5.75
CostAggOcc(WLS) [57]	1.96	1.13	11.9	8.57	5.89
GeoSup [34]	1.83	0.26	13.2	8.89	6.05
RecursiveBF [82]	2.51	0.88	12.1	8.91	6.10
InfoPermeable [17]	1.53	0.88	13	9.16	6.14
HierarchicalStereo [36]	1.87	0.75	10.98	11.36	6.24
DistincSM [88]	1.75	0.69	13	9.91	6.34
RegionTreeDP [43]	1.64	0.57	11.9	11.9	6.50
AdapWeight [87]	1.85	1.19	13.3	9.79	6.53
HistoAggr1 [55]	2.71	0.97	13.08	9.47	6.56
Proposed(NGN)	4.3	1.35	10.7	10.5	6.71
SemiGlob [31]	3.96	1.57	12.2	9.75	6.87
FastBilateral [53]	2.8	0.92	15.3	9.31	7.08
Proposed(REAF)	5.55	2.41	11.4	11.8	7.79
VariableCross [91]	2.65	0.96	15.1	12.7	7.85
Proposed(GSF)	4.89	4.59	12.9	10.9	8.32
ScaleSelectCTF [37]	4.05	2.49	12.29	16.96	8.95
Proposed(MG)	5.38	3.82	15.6	16.3	10.28
Acc.Bound.CTF [66]	11.5	5.22	13.7	10.8	10.31
Proposed(NN)	5.38	3.85	15.9	16.5	10.41
TreeDP [76]	2.84	2.1	23.9	18.3	11.79

sponds to 2.91% of the total pixels in the image. When compared to the second best method which has 42% more erroneous estimates relative to the top performer, the proposed method has 24% more erroneous estimates. According to the recursive edge aware filtering approach [17] which is the baseline for the proposed approach, the relative error percentage is 9%. This shows that the pro-

Table3.2: Disparity estimation accuracy comparisons of the selected methods for non-occluded pixels in Middlebury evaluation data set.

Method	**Tsukuba**	**Venus**	**Teddy**	**Cones**	**Avg. % Bad Pixels**
PatchMatchFilter [50]	1.74	0.33	2.52	2.13	1.68
InfoPermeable [17]	1.06	0.32	5.6	2.65	2.41
DomainTfm [60]	1.75	0.24	5.7	2.49	2.55
SegmentTree [54]	1.25	0.2	6	2.77	2.56
HistoAggr2 [56]	1.93	0.16	5.88	2.41	2.60
CrossLMF [49]	2.46	0.27	5.5	2.34	2.64
CostFilter [35]	1.51	0.2	6.16	2.71	2.65
NonLocal [81]	1.47	0.25	6.01	2.87	2.65
P-LinearS [21]	1.1	0.53	6.69	2.6	2.73
GeoDiff [22]	1.88	0.38	5.99	2.84	2.77
RecursiveBF [82]	1.85	0.35	6.28	2.8	2.82
GeoSup [34]	1.45	0.14	6.88	2.94	2.85
CostAggOcc(WLS) [57]	1.38	0.44	6.8	3.6	3.06
DistincSM [88]	1.21	0.35	7.45	3.91	3.23
SemiGlob [31]	3.26	1	6.02	3.06	3.34
AdapWeight [87]	1.38	0.71	7.88	3.97	3.49
Proposed(NGN)	**3.52**	**1**	**5.74**	**4.37**	**3.66**
HierarchicalStereo [36]	1.61	0.36	8.48	4.65	3.78
RegionTreeDP [43]	1.39	0.22	7.42	6.31	3.84
HistoAggr1 [55]	2.47	0.74	8.31	3.86	3.85
FastBilateral [53]	2.38	0.34	9.83	3.1	3.91
Proposed(REAF)	**4.44**	**1.74**	**5.47**	**4.75**	**4.10**
ScaleSelectCTF [37]	2.37	1.5	4.44	10.19	4.63
VariableCross [91]	1.99	0.62	9.75	6.28	4.66
Proposed(GSF)	**3.87**	**3.72**	**6.78**	**4.83**	**4.80**
Proposed(MG)	**3.76**	**2.55**	**7.44**	**6.6**	**5.09**
Proposed(NN)	**3.78**	**2.57**	**7.43**	**6.98**	**5.19**
Acc.Bound.CTF [66]	10.2	4.58	8.39	5.03	7.05
TreeDP [76]	1.99	1.41	15.9	10	7.33

posed method performs close to the state-of-the-art local methods. Moreover, the proposed method also has a comparable performance to that of semi-global methods. The region based semi-global approach [43] performs better than the proposed method with a marginal increase in estimation accuracy. On the other hand, the proposed method has a slightly better estimation accuracy compared to other semi-global methods. Among the CTF methods [36, 37, 66], the proposed approach ranks as the second best performer with a close performance to the top performer.

It is important to note that the proposed method does not have a post processing step to refine disparity estimates or to handle occlusions unlike most of the local methods. Thanks to proposed disparity estimate transfer method, the occlusions are inherently handled and the evaluation results of *Teddy* and *Cones* stereo image pairs indicate the success of the method, since these stereo pairs involve large occluded regions. The percentage of the erroneous estimates for these stereo pairs are fewer than most of the compared methods. Especially, for *Teddy* pair, the proposed method is the second best performer. Table 3.2 provides the evaluation results at the non-occluded regions. It can be easily observed from Table 3.2 that the proposed information transfer method (NGN), provides performance increase over the other methods for disparity transfer from coarse scale to fine scale (REAF,GSF,MG,NN).

For the sake of visual interpretation, the disparity estimation results and erroneous estimates for the left images of the four stereo image pairs are provided in Fig. 3.26 together with the left images of the four image pairs and corresponding ground truth disparity maps.

The evaluation result of the proposed method for the extended data set is provided in Table 3.3. The results of the methods using state-of-the-art EAF techniques are included in this table. These results are retrieved from [16]. In [16], the comparative tests for the listed techniques in Table 3.3 are conducted for the extended data set by only altering the cost aggregation step of the disparity estimation algorithmic blocks. In other words, the same matching cost function and occlusion handling method are utilized for all tests and only the cost aggregation step is altered by using different EAF strategies. According to the results provided in [16], *Information Permeability* (IP) based filtering outperforms the other EAF techniques.

In Table 3.3, it can be observed that the proposed method performs better than all the other EAF based methods except for IP based method. This is an

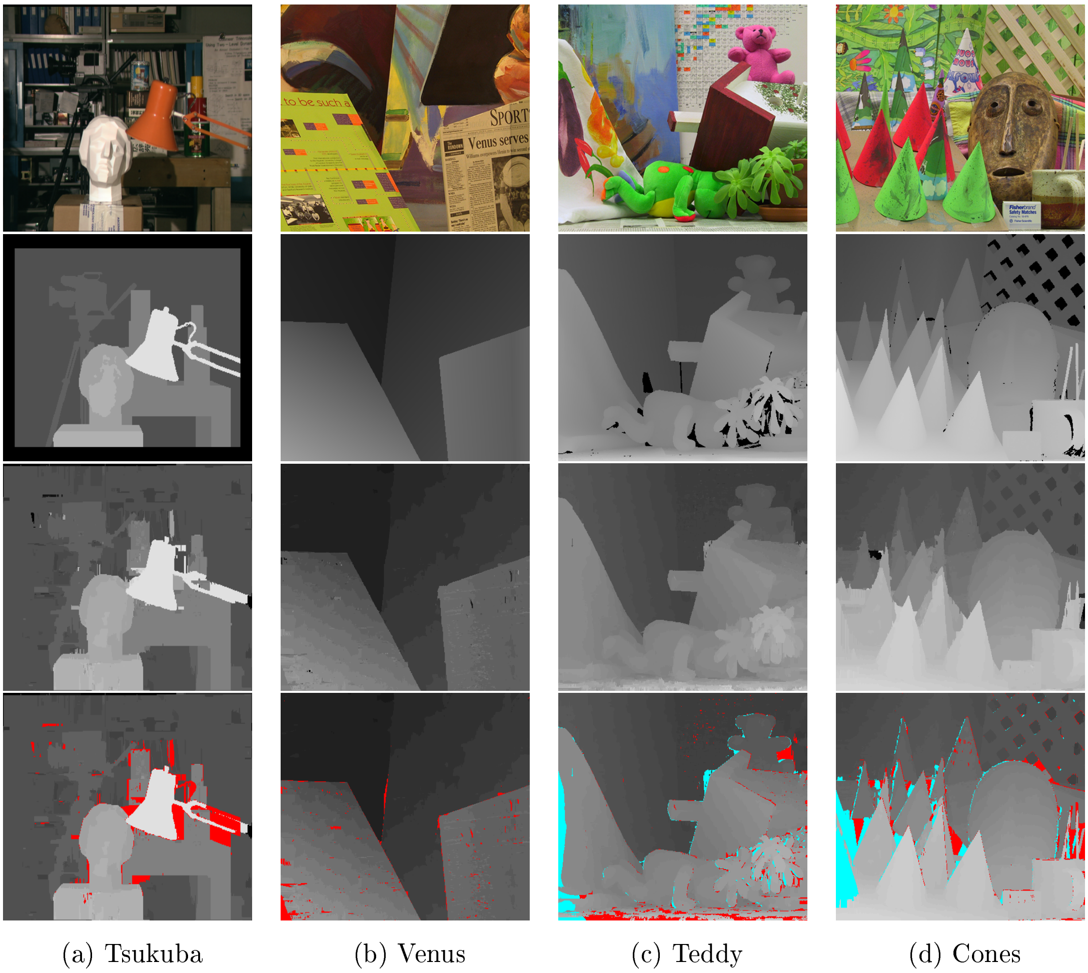

(a) Tsukuba (b) Venus (c) Teddy (d) Cones

Figure 3.26: First row: Left images from Middlebury evaluation data set [62], Second row: Corresponding ground truth disparity maps, Third row: Disparity maps, D^{visual} (according to Eq. 3.76), from the proposed algorithm using optimization based disparity estimate transfer (Proposed(NGN)), Fourth row: Disparity estimation errors for larger than 1 disparity level where blue regions represents the errors at occluded regions, red regions represents the errors at non-occluded regions. The coordinate ranges and the parameters according to Fig. 3.25 are explained in Sec. 3.8.1.

important result, since the proposed method aims complexity reduction by utilizing recursive edge aware filtering exploited in IP [16] and this result indicates that the performance sacrifice to gain from computation is marginal. As it is also discussed in the following parts, the proposed method estimates disparities much more faster than recursive EAF based methods and decreases the estimation performance by only 2% in return and still performs better than all the state-of-the-art EAF based methods.

Table3.3: Disparity estimation accuracy comparisons of the EAF based methods for all pixels in the extended data set.

Method	**Avg. % Bad Pixels ($\Delta d > 1$)**	**Avg. % Bad Pixels($\Delta d > 2$)**
InfoPermeable [17]	14.20	10.30
Proposed(NGN)	**14.67**	**11.51**
Guided [30]	15.10	11.80
Proposed(REAF)	**16.33**	**12.25**
Bilateral [74]	16.90	13.10
Var. Cross [91]	17.10	12.60
2-pass Bilateral [58]	17.20	13.00
GeoSup [34]	17.50	12.70
Adapt. Box [75]	18.20	13.50
O(1) Bilateral [83]	18.40	14.10
Proposed(GSF)	**18.90**	**13.25**
Cos. Int. [24]	19.40	14.00
Proposed(MG)	**19.68**	**17.35**
Proposed(NN)	**20.25**	**17.30**

Additionally, the superior performance of the proposed method in the extended data set is a good indicator of the robustness of the method. The extended data set contains stereo image pairs from various scenes including thin and thick objects, highly textured, weakly textured, occluded and non-occluded regions with different scene brightnesses as shown in Figs. 3.34 - 3.41. Most of the methods performing better in Middlebury evaluation data set perform worse than the proposed method in the extended data set. This is mainly because the predefined window size based aggregation methods cannot properly handle untextured regions. It should be noted again that the proposed method provides the estimations without any post-processing or explicit occlusion handling step.

The disparity estimation results for the extended data set is provided in Figs. 3.34 - 3.37 and in Figs. 3.38 - 3.41 separately in order to illustrate the strengths and weaknesses of the proposed method, respectively. It can be observed from Figs. 3.40, 3.41 that CTF scheme of the proposed method causes poor estimates at the fine details. On the other hand, unless most of the scene is composed of fine structures, the proposed method can estimate disparities accurately regardless of the presence of a texture and the proposed method is robust to different scene brightnesses as shown in Figs. 3.34 - 3.37.

3.8.3 Time Complexity

The time complexity of the proposed method is compared with the state-of-the-art local stereo matching methods included in the accuracy comparison in the previous part. The execution times of the methods are available in the corresponding studies [16,21,36,37,49,50,54,56,57,60,81,87]. These execution times are environment dependent and are not suitable for a fair comparison. Therefore, these execution times should be standardized. Thanks to the execution time comparisons provided in the studies [16,21,36,37,49,50,54,56,57,60,81,87], the relative time complexities of the methods can be determined. In other words, by using the execution time results of the several methods in the same environment, how much times a method is faster than another method can be determined.

Prior to determination of relative execution times of the methods, a consistency check is performed first. The comparison results of different studies are checked to verify whether the provided results support each other. Namely, if *Geodesic Support Filtering* based method is 10 times slower than *Bilateral Filtering* based method according to the implementation environment used in one study, then it is expected to observe similar result according to the implementation environment used in the other study. The studies which have consistent results among each other [21,36,37,49,50,54,57,60,81,87] are used to determine relative execution times of the methods.

To provide a baseline for the standardization, currently the fastest stereo matching methods [17,60] are implemented and their relative execution times according to the execution time of the proposed method is determined. Using these results, the execution times of the other methods relative to the proposed method is retrieved.

In Fig. 3.27, a joint comparison of disparity estimation accuracy and execution time of the methods are provided. The estimation accuracies are represented by the average percentage of the erroneous pixels that the methods have for the

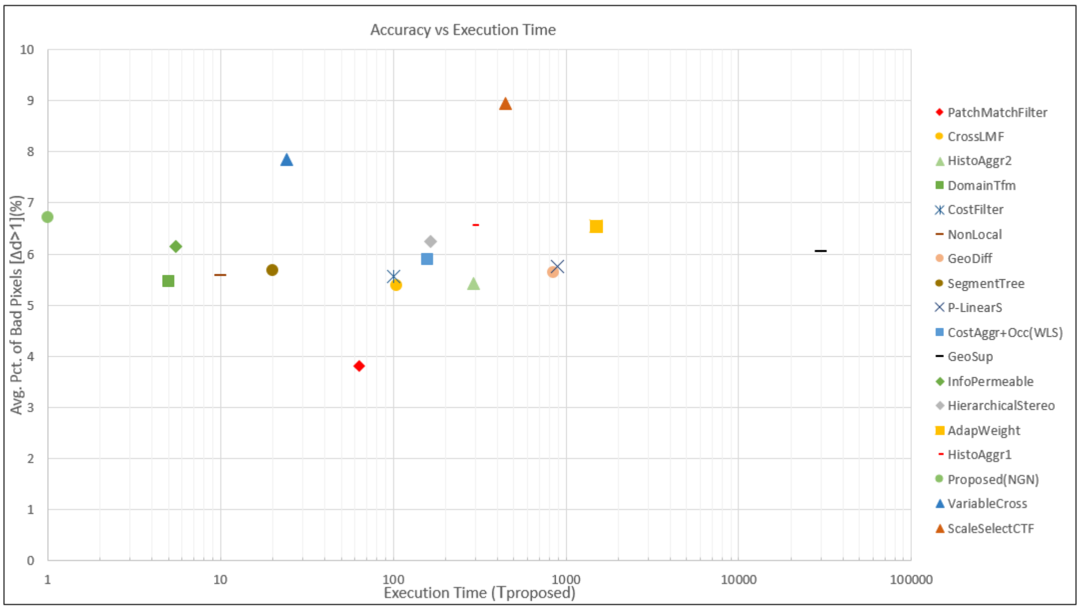


Figure 3.27: Joint comparison of disparity estimation accuracy and execution time of the methods listed in Table 3.1. The execution times are standardized according to the execution time of the proposed method.

four stereo image pairs used in Middlebury evaluation. The execution times are provided relative to the execution time of the proposed method. That is to say, if the execution time of a method is 5, then it means that method is 5 times slower than the proposed method. Consequently, it is better to be in the bottom-left part of the plot in Fig. 3.27. The execution times are determined according to a problem with image size of 665 × 555 and 100 disparities in the disparity search range; since, the previous studies [21,36,37,49,50,54,57,60,81,87] provide execution times for this problem size.

It can be observed from Fig. 3.27 that the proposed method outperforms the state-of-art methods in terms of execution time with marginal decrease in performance. The proposed approach is almost 60 times faster than the top performing *Patch Match Filter* [50] based method and 5 times faster than the recursive EAF based methods which are reported to be currently the fastest stereo matching methods for these comparison tests. On the other hand, the performance drop is 1.77 times more erroneous pixels than the top performer and only 1.09 times more erroneous pixels than the fastest methods. It is important to note that these gaps in the execution time increase as the problem size increases, since,

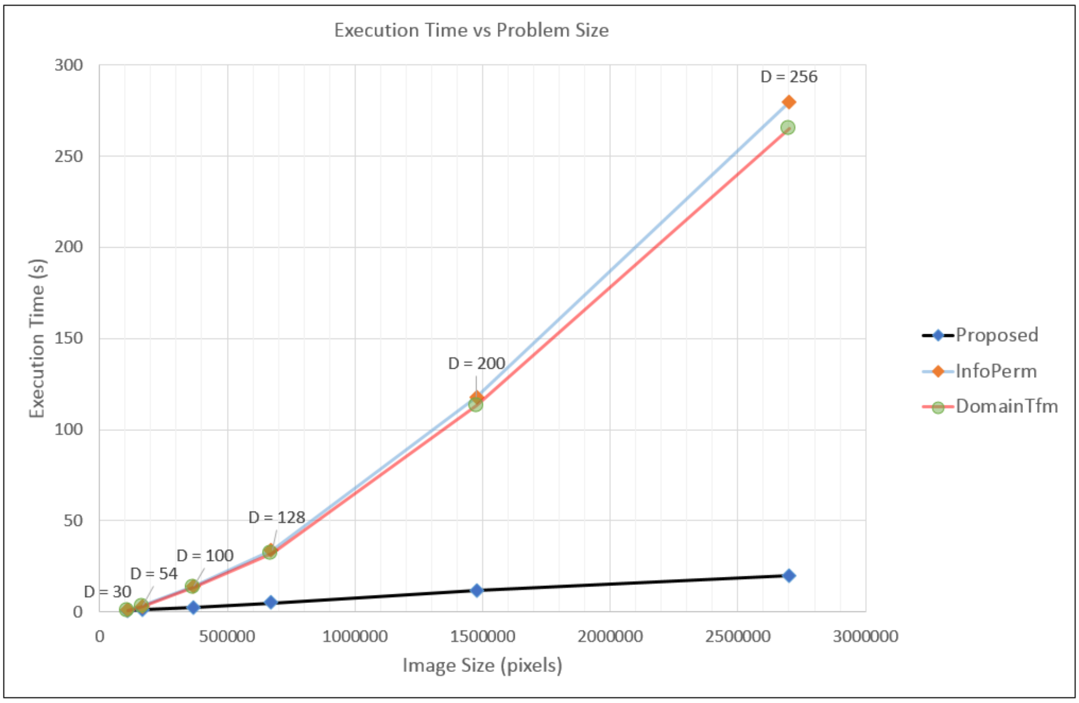


Figure 3.28: Comparison of the execution time of the proposed method with the execution times of the fastest stereo matching methods in the literature [17, 60] with different problem sizes. The disparity search range information is attached on the plot for each image size.

the proposed method does not introduce a complexity factor related with the disparity search range. This behaviour is illustrated in Fig. 3.28 by plotting the execution times of the fastest rival methods with respect to the problem size. It can be observed that the execution time of the proposed method increases linearly with the image size and is not affected from the disparity search range. Therefore, as the problem size increases, the proposed method becomes relatively much faster than all the previous methods. Moreover, the accuracy sacrificed in return is marginal and the quality of the disparity estimation results are competitive. Visual results provided in Figs. 3.34 - 3.41 support these claims for those test images. Finally, when compared to CTF methods which aim to decrease computational complexity by reducing the search space, the proposed methods outperforms all of the CTF methods with at least 160 times more reduced computation time and among those methods, only one of them (*HierarchicalStereo* [36]) has less erroneous estimates than the proposed method, yet it has only 7% less than the proposed method, which corresponds to 0.47%

of all pixels in an image.

3.8.4 Analysis of the Proposed Method

The proposed method is analysed in two parts regarding to the results of the experiments. First part discusses the strong and weak sides of the method and the second part analyses the characteristics of the method under different parameters.

3.8.4.1 Analysis of the Disparity Estimation

The proposed method provides very fast disparity estimation from the stereo image pairs and solves occlusion problem inherently, and this enables further reduction in complexity. The disparity estimations at the regions where there is no fine structures are quite satisfactory. The disparity estimation results of such scenes are provided in Figs. 3.34 - 3.37. It can be observed that the proposed method is not affected from texture characteristics of the regions. On the other hand, the CFT scheme of the proposed method decreases the quality of the disparity estimations especially for the scenes that have fine structures as shown in Figs. 3.40 and 3.41. Yet, when the gain from the computational complexity is considered, even the poor results illustrated in Figs. 3.40 and 3.41 might be admissible if time is a concern.

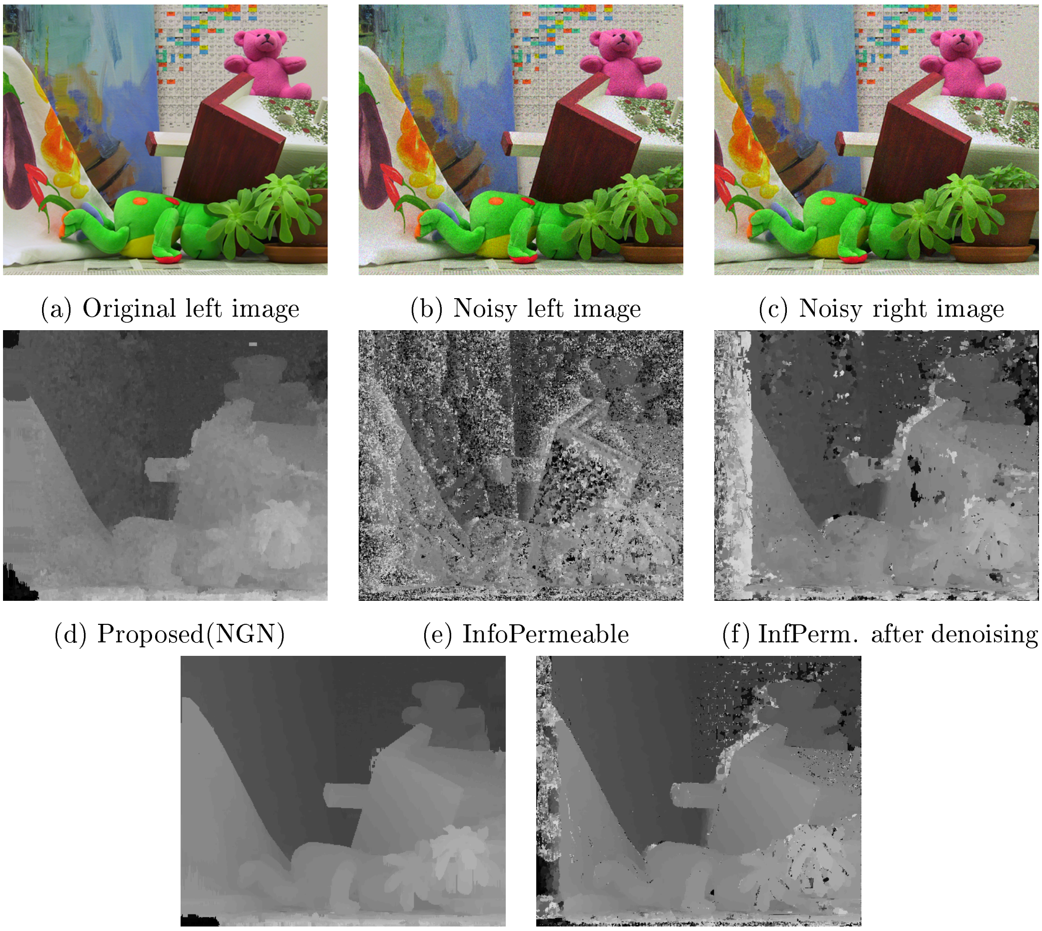
The linear time complexity of the proposed method is validated by the experiments and it is illustrated in Fig. 3.28. With its linear time complexity, the proposed method becomes the faster alternative of the current state-of-the-art fastest methods that utilize recursive EAF. Moreover, though the proposed method exploits the recursive EAF, it does not have the weaknesses of the recursive EAF based methods in the presence of high texture or noise. The color variations in the highly textured regions or the noisy regions prevent data passing among pixels and this results in poor estimates due to lack of support from the neighbouring pixels. The CFT scheme of the proposed method implicitly handles those regions thanks to the low pass filtering during the pyramid generation. In Fig. 3.29, the disparity estimation results of the proposed method and the recursive EAF based method (IP [17]) is provided in order to interpret the weaknesses of the recursive EAF based methods in the presence of noise and texture. It can be concluded that the recursive EAF based methods cannot handle noisy data unless they have a preprocessing denoising step which introduces additional computation. In addition to this, they require post processing to refine the disparity estimates at the textured regions.

3.8.4.2 Effects of Parameters and Different Information Transfer Methods

The proposed method has two significant parameters that affect both the execution time and the disparity estimation performance. These parameters are the number of disparity candidates and the initial disparity search range, or namely, the maximum pyramid level. In order to examine the characteristics of the proposed method in terms of these parameters, tests are conducted on the extended data set and both execution time and disparity estimation performance of the method are analysed under different parameters.

Firstly, the effect of the number of disparity candidates on the estimation performance and computation time is analysed. Increasing the number of disparity candidates increases the execution time due increase of problem size as shown in Fig. 3.30a. Yet, increasing the number of disparity candidates does improve the performance up to some point as shown in Fig. 3.30b. The reason of this is the spurious matches occur especially in the textured regions when the search space is enlarged. Therefore, keeping the number of disparity candidates small is beneficial for both estimation performance and faster execution time.

Secondly, the effect of the initial disparity search range is analysed. As discussed in Sec. 3.7.3, there is an optimal initial search range for the optimal execution time. This initial search range implies the maximum pyramid level; in other words, how much the image should be downsampled. Downsampling the image too much causes thin objects to be vanished in the coarse images and prevents their disparities from being estimated. In Sec. 3.7.3, it is shown that near op-



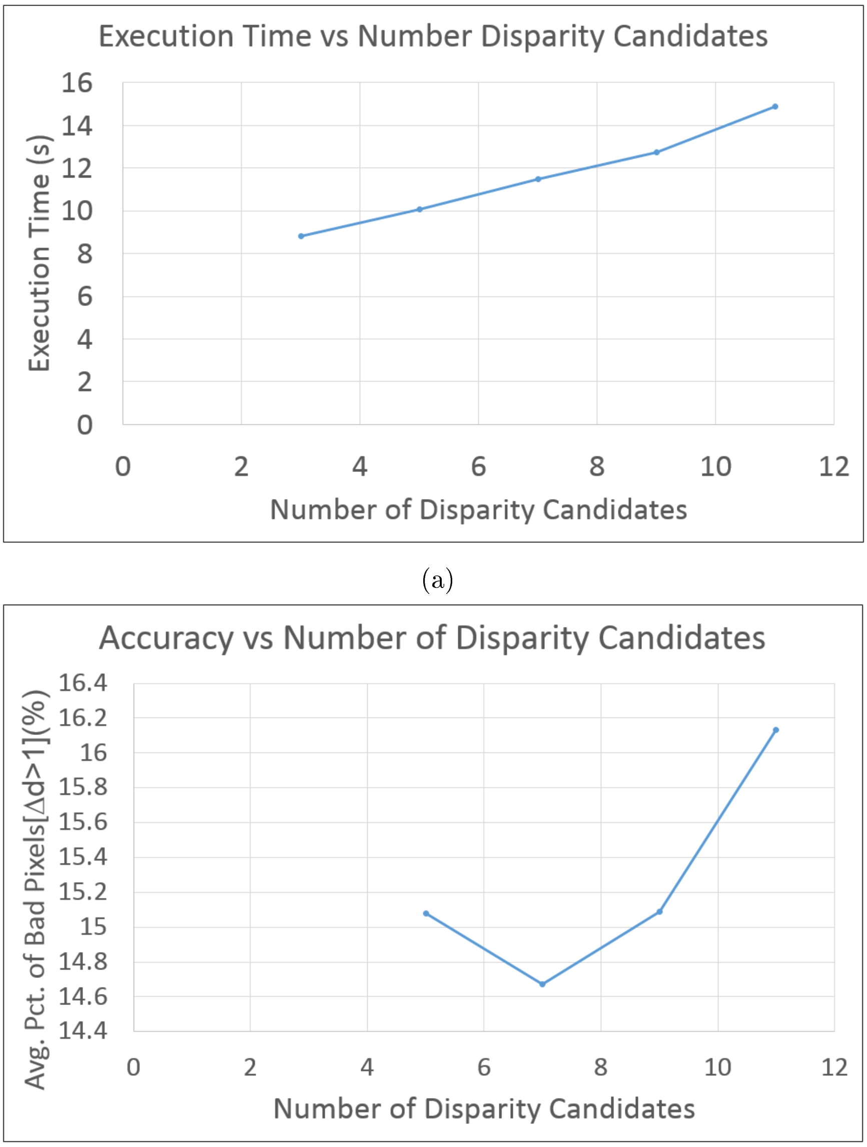
(a) Original left image (b) Noisy left image (c) Noisy right image

(d) Proposed(NGN) (e) InfoPermeable (f) InfPerm. after denoising

(g) Proposed(NGN) (h) InfoPermeable

Figure 3.29: Behaviours of the recursive EAF based algorithms in the presence of noise (d)-(f) and texture (g), (h). (a): Original left image from *Teddy* stereo pair [62]. (b)-(c): Left and right images with additive Gaussian noise. (d): Disparity estimation result, D^{visual} (Eq. 3.76), of the proposed method for noisy pairs. (e): Disparity estimation result of the *Information Permeability* [17] based method for noisy pairs. (f): Disparity estimation result of the *Information Permeability* [17] based method for noisy pairs after denoising. (g): Disparity estimation result, D^{visual} (Eq. 3.76), of the proposed method for the original stereo pair. (h): Disparity estimation result of the *Information Permeability* based method for the original stereo pair. It can be observed that proposed method has superior performance in the presence of noise and texture. The coordinate ranges and the parameters are the same as in Fig. 3.25.

timal initial search ranges do not break the time complexity of the method and consequently, do not degrade the efficiency, thanks to the piecewise constant behaviour of the complexity function in Eq. 3.70 near the optimal point. This claim is validated by the experiments. Fig. 3.31a shows that the small devia-

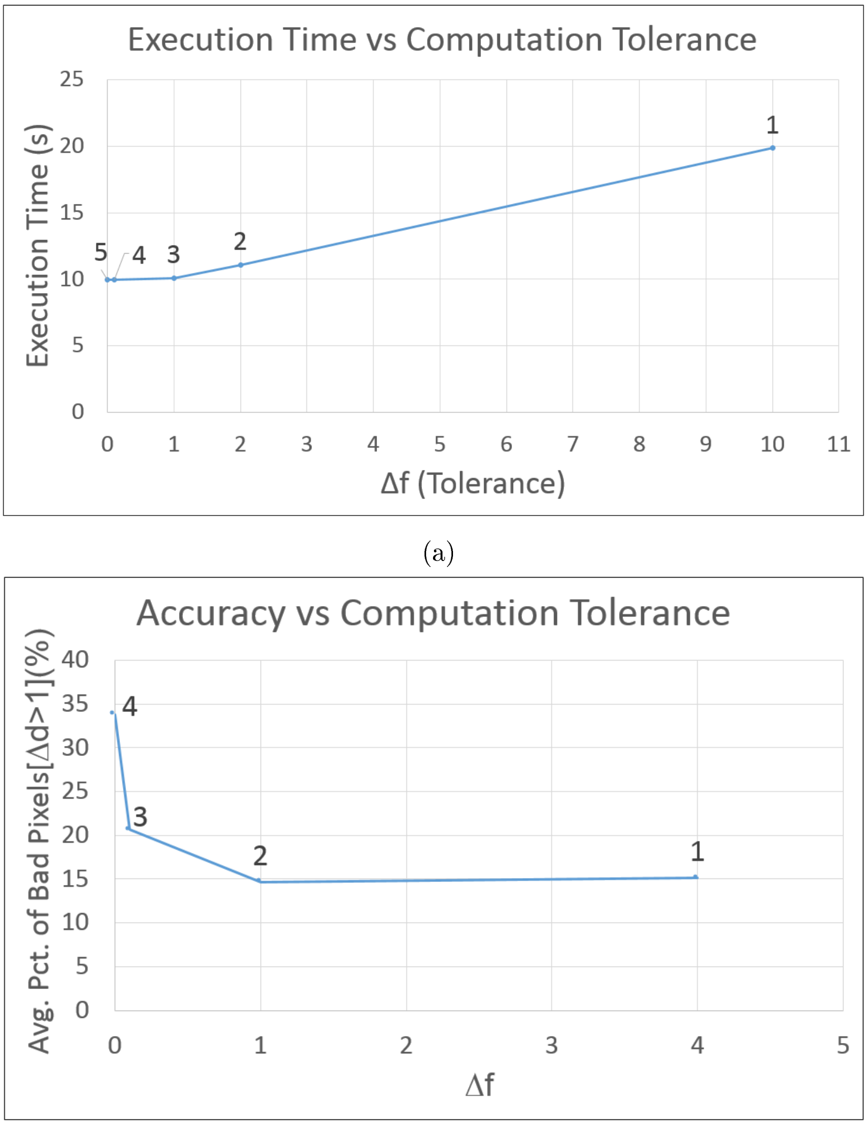


(a)

(b)

Figure 3.30: Effect of increasing the size of the disparity candidate set on execution time (a) and on disparity estimation accuracy (b).

tions from optimal point introduces negligible additional execution time. This means, fewer pyramid levels has almost the same computational complexity as the optimal number of pyramid levels. However, fewer pyramid levels improves the estimation performance considerably as shown in Fig. 3.31b. The tolerance parameter, which is defined by Eq. 3.74, in these figures denotes the allowed increase in the value of the complexity function derived in Eq. 3.70 when a different pyramid level is to be used as the coarsest scale instead of the optimal level. In other words, referring to Fig. 3.31a, if the tolerance is set to 1, then

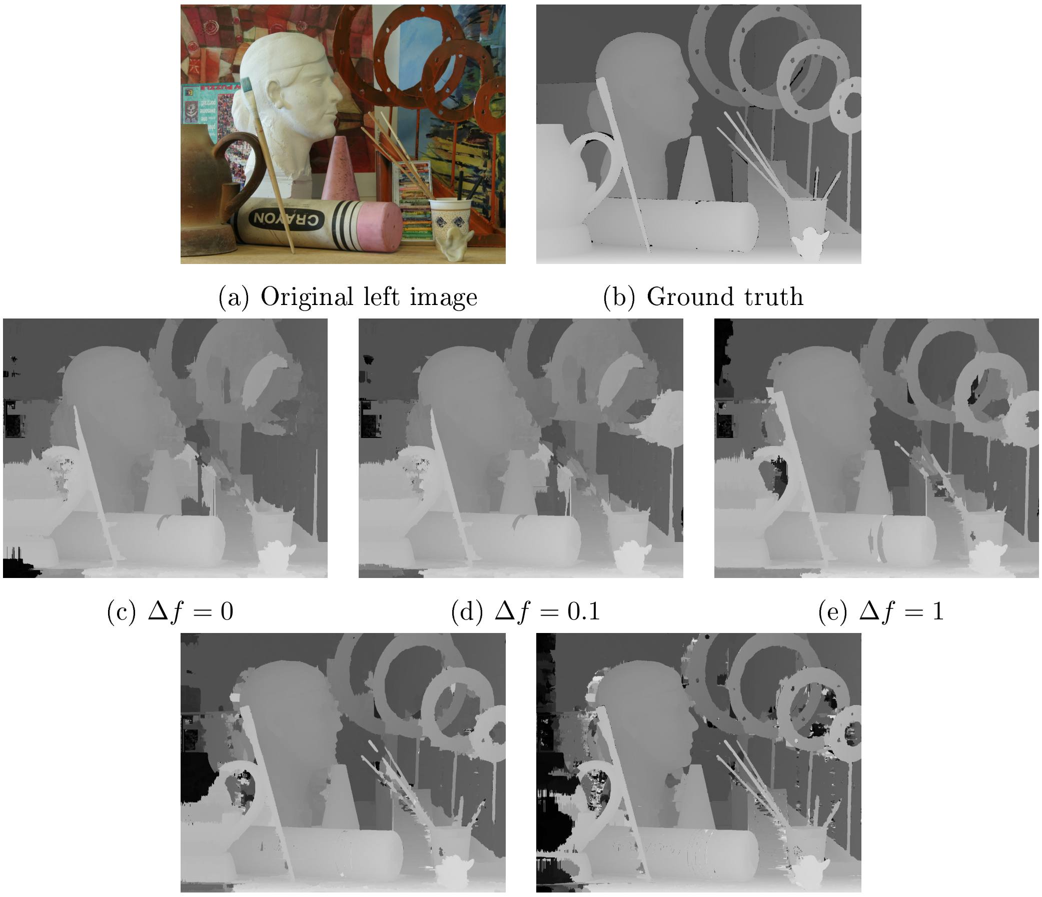


(a)

(b)

Figure 3.31: Effect of different tolerances on the execution time (a) and estimation performance (b). The tolerance is the allowed increase in the value of the complexity function in Eq. 3.74. Larger tolerances result fewer pyramid levels and better preservation of fine structures. The pyramid levels corresponding to the tolerance values are attached on the plot at the tolerance value samples.

the maximum pyramid level cannot be smaller than 3; because, smaller pyramid levels increase the value of the function in Eq. 3.70 more than tolerated value. For the sake of visual interpretation, the disparity estimation results corresponding different tolerance parameters are provided in Fig. 3.32. It can be observed that the quality of the disparity maps increases without introducing significant complexity provided that the tolerance is kept small.



(a) Original left image

(b) Ground truth

(c) $\Delta f = 0$

(d) $\Delta f = 0.1$

(e) $\Delta f = 1$

(f) $\Delta f = 2$

(g) $\Delta f = 10$

Figure 3.32: Effect of different tolerances on the estimation performance for the *Art* stereo pair [62]. The tolerance is the allowed increase in the value of the complexity function in Eq. 3.74. Larger tolerances result fewer pyramid levels and better preservation of fine structures. The results, $D^{visual}[x, y]$, in (c-g) is obtained according to Eq. 3.76 by using proposed disparity transfer method and the coordinate ranges and the parameters are explained in Sec. 3.8.1.

Finally, the computational complexities and the disparity estimation performances of using different disparity estimate transfer methods are analysed. In Fig. 3.33, the joint plot of the disparity estimation performances for the extended data set and execution times of the disparity transfer methods is given. The execution times are given relative to fastest approach. The nearest neighbour (NN) approach is the baseline, since it is the fastest approach. It can be observed that the proposed disparity transfer method is the top performer and only 1.17 times slower than the NN approach but has 38% less erroneous pixels.

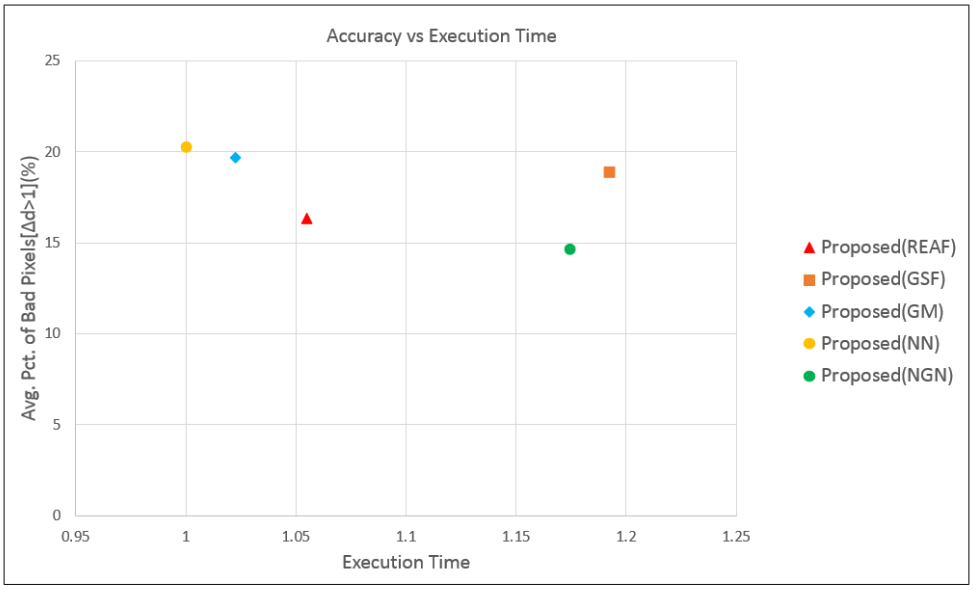
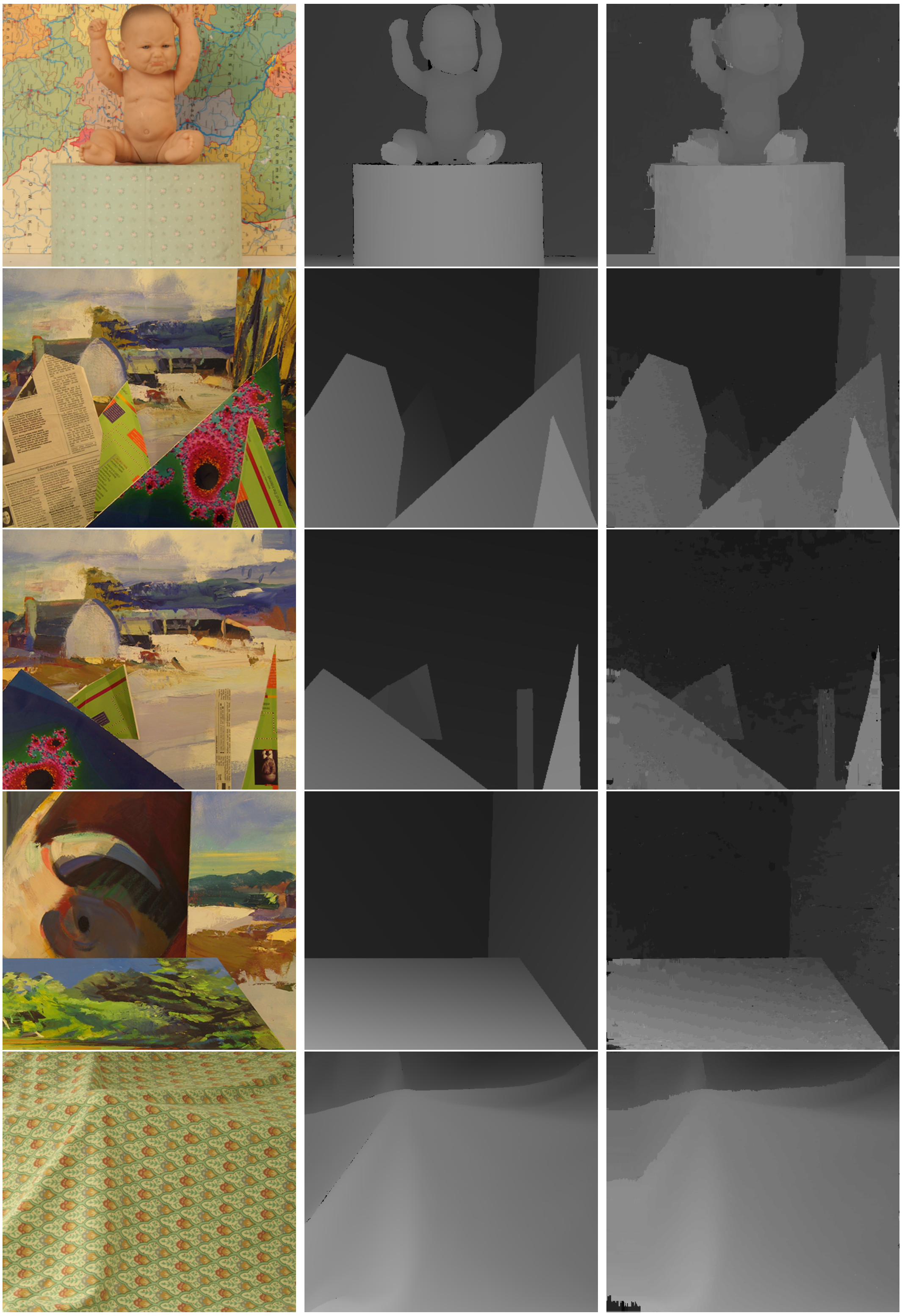
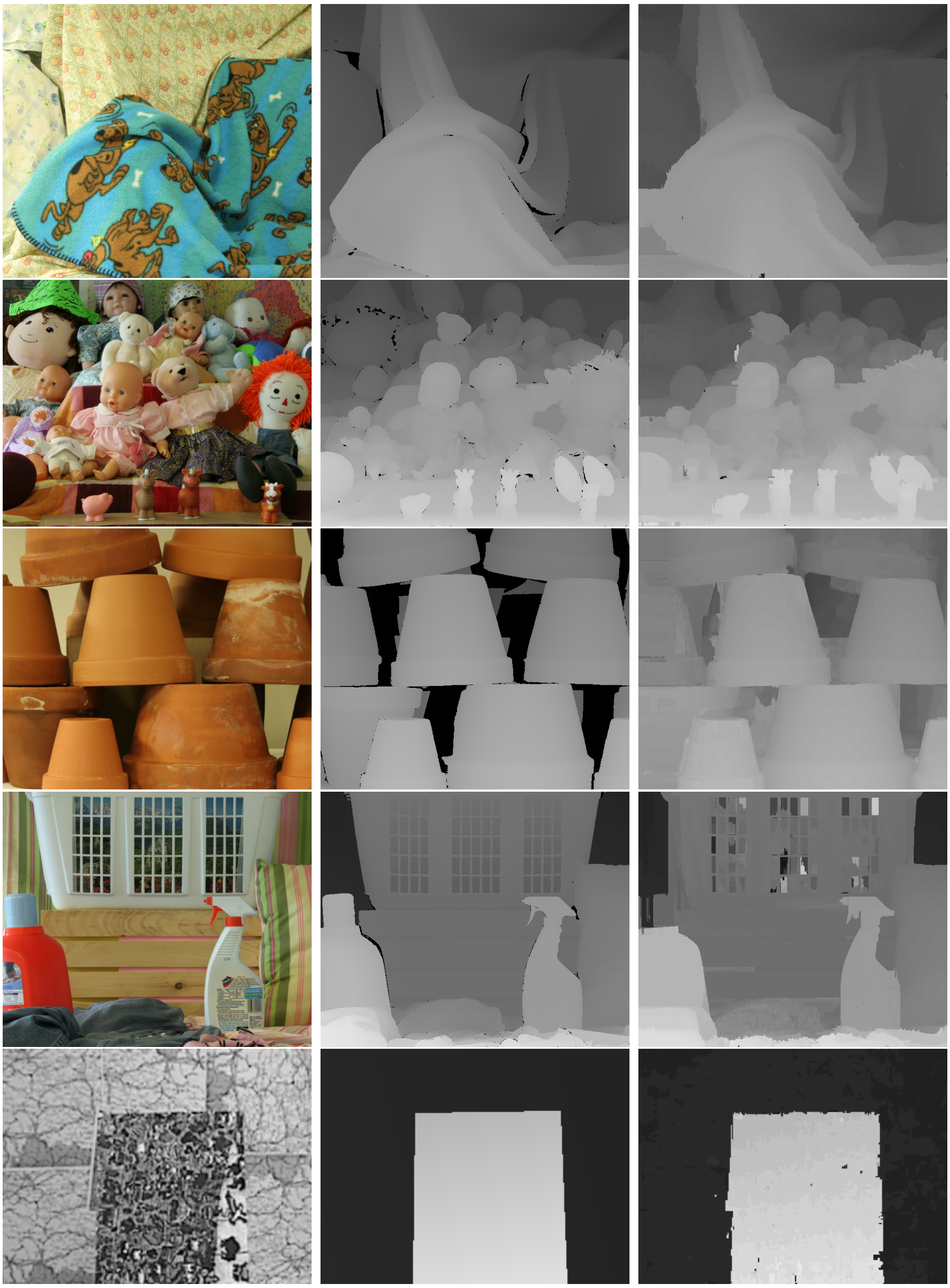


Figure 3.33: Joint comparison of the disparity estimation accuracy and execution time of utilizing different disparity information transfer methods. The execution times are standardized according to the execution time of the fastest method. (NN): Nearest neighbour. (MG): Nearest neighbour by considering goodness of the matching costs. (REAF): Recursive edge-aware filtering based method. (GSF): Geodesic support filtering based method. (NGN): Proposed optimization based nearest geodesic neighbour method.



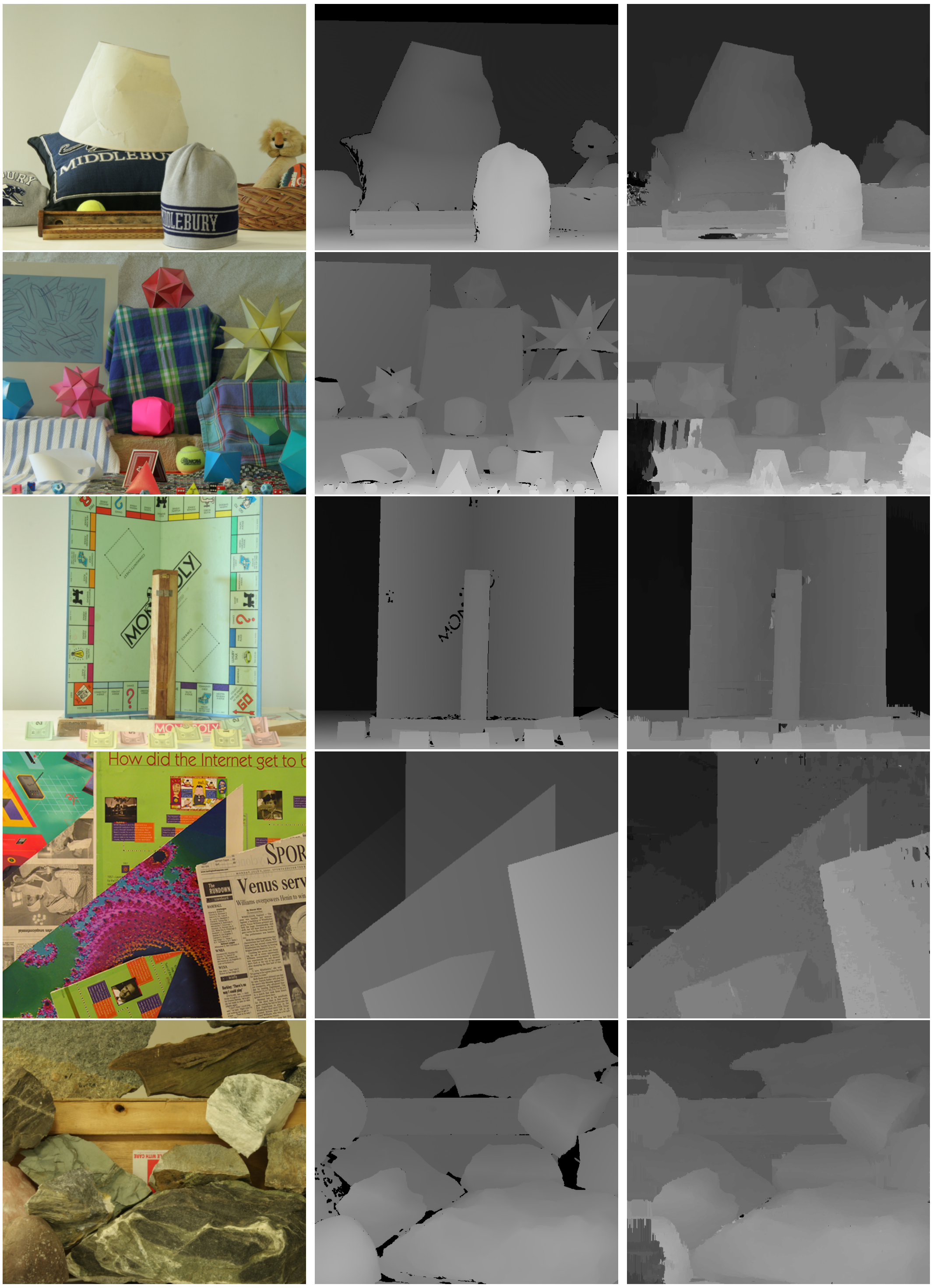
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.34: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs well.



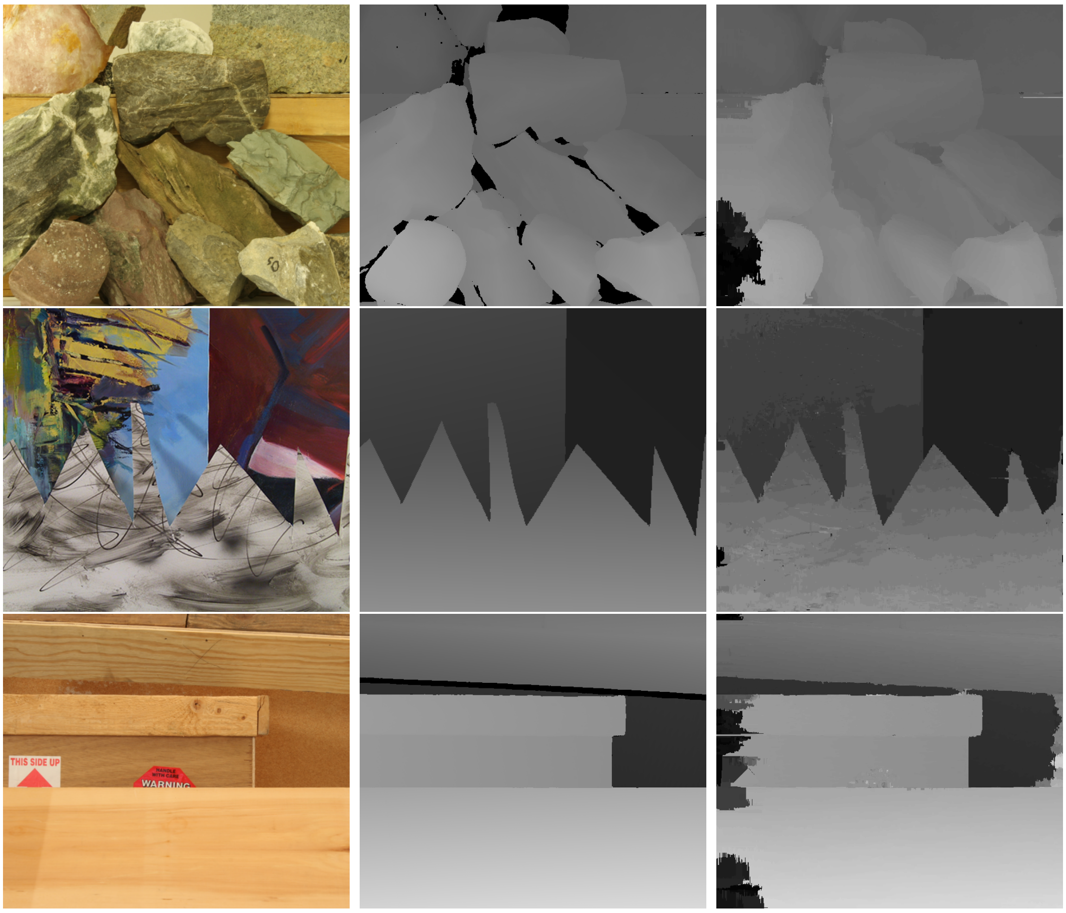
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.35: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs well.



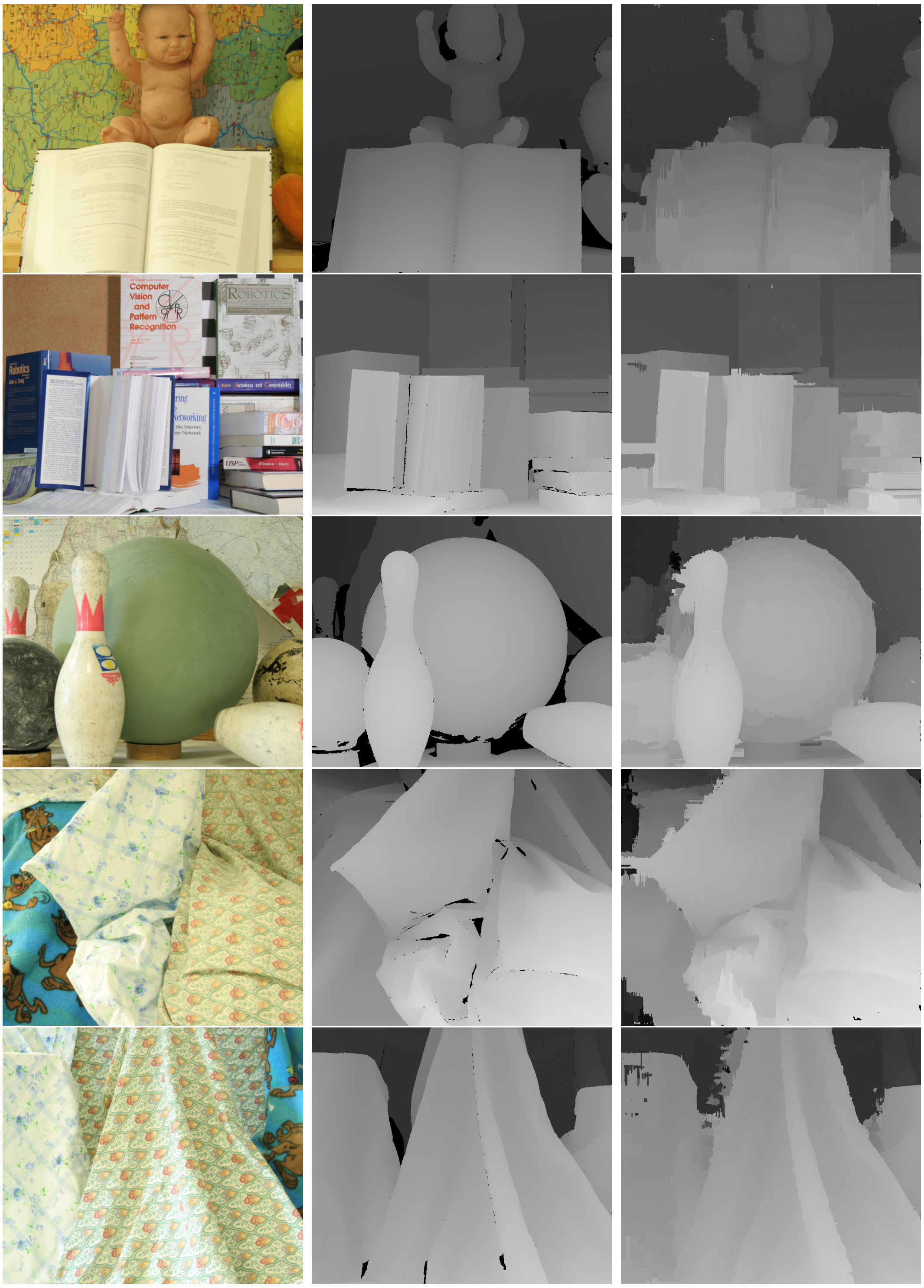
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.36: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs well.



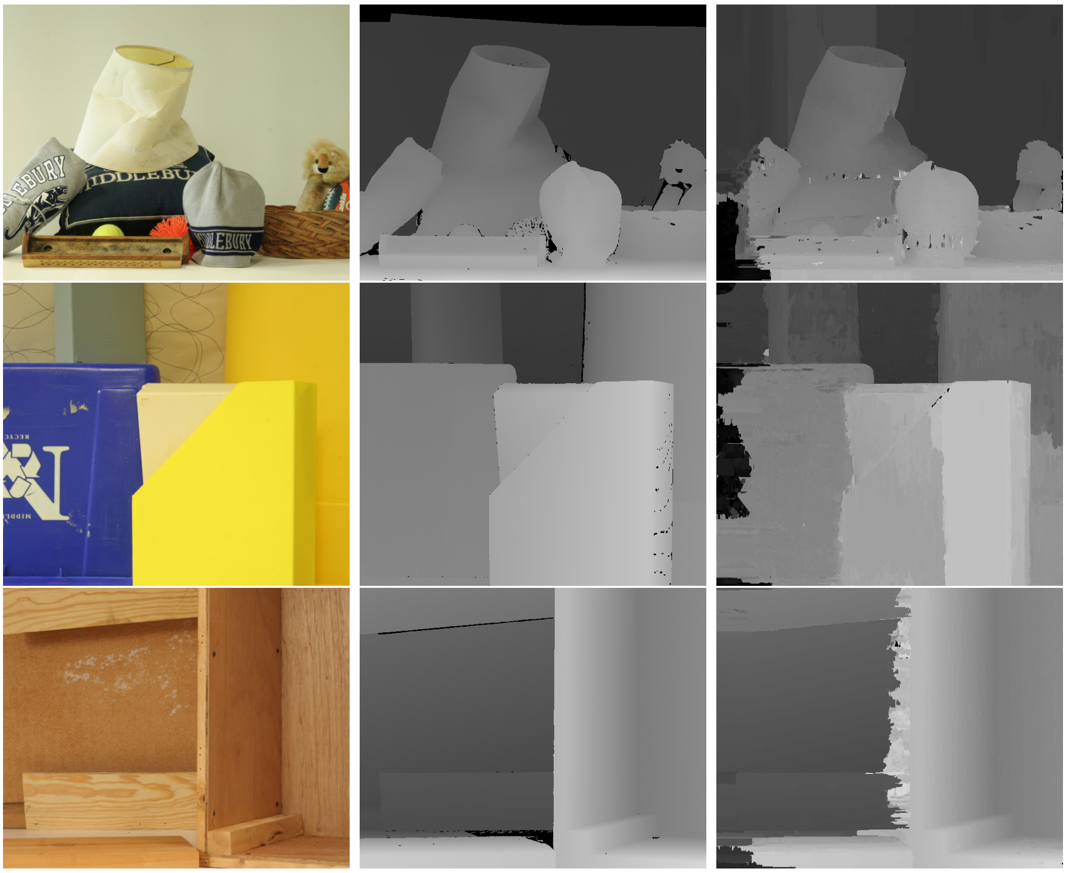
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.37: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs well.



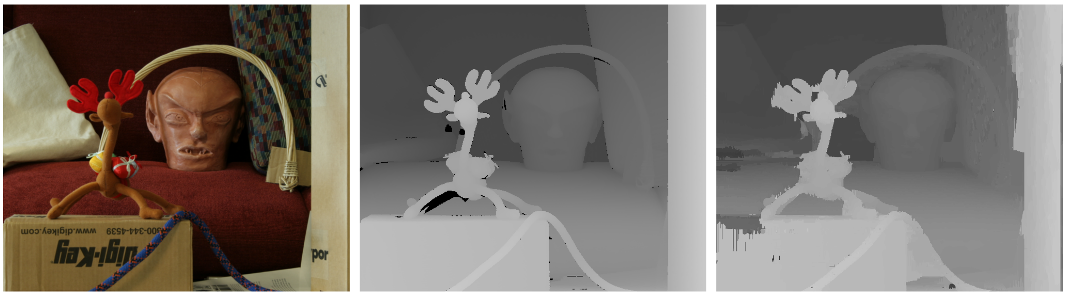
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.38: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs on the average.



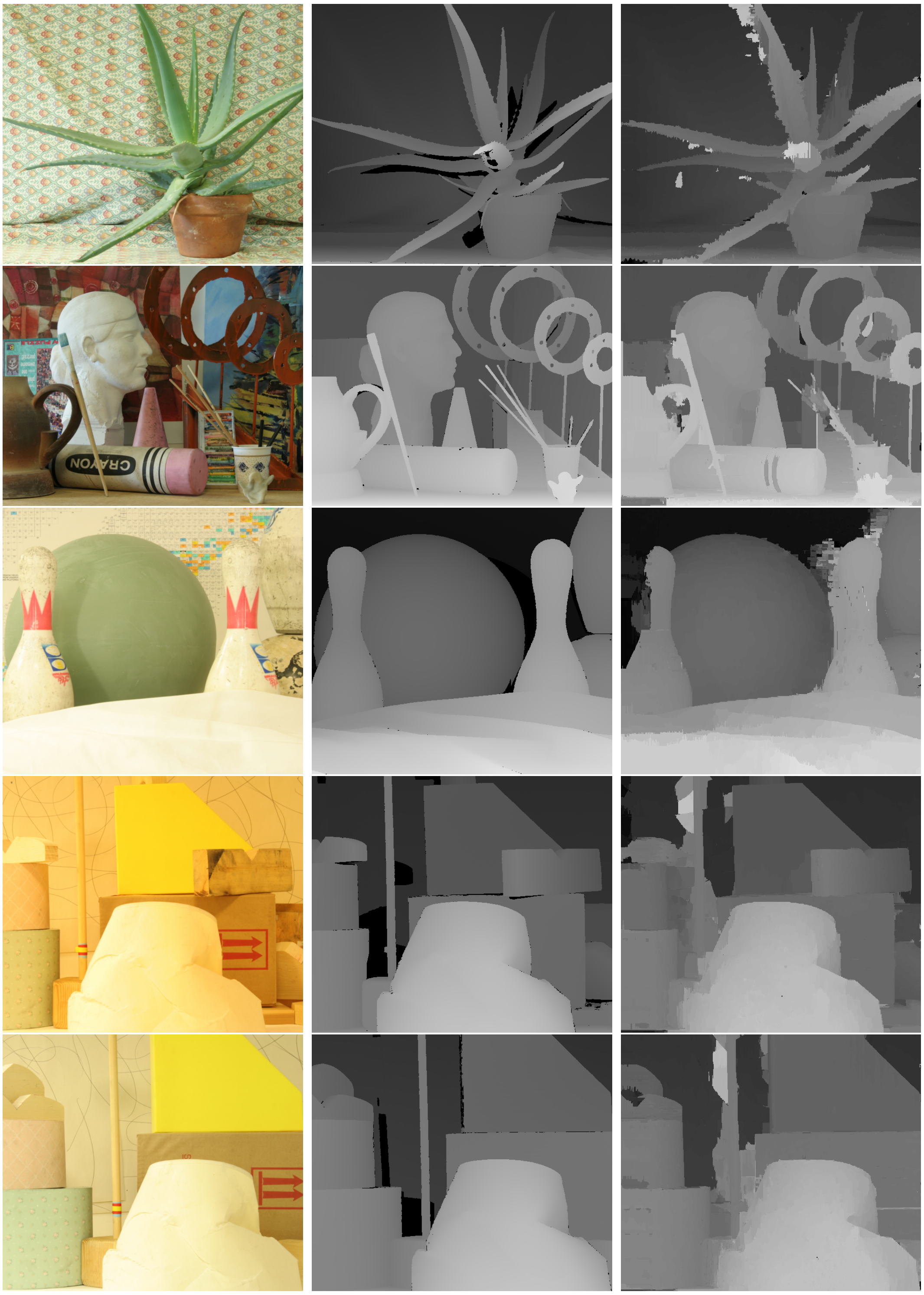
(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.39: Disparity estimation results, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method performs on the average.



(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.40: Disparity estimation result, $D^{visual}[x, y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method is weak.



(a) Original left image (b) Ground truth (c) Estimation results

Figure 3.41: Disparity estimation results, $D^{visual}[x,y]$ (Eq. 3.76), of the proposed method for the extended Middlebury dataset [62]. The coordinate ranges and the parameters are explained in Sec. 3.8.1. These are the scenes where the proposed method is weak.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

4.1 Summary

Within the scope of this thesis study, the problem of reducing the computational complexities of local stereo matching methods on both cost aggregation and correspondence search step is addressed. Two main perspectives of the problem are attacked. The first one is to perform efficient cost aggregation for the methods aiming complexity reduction on disparity search range. This is a challenging task; since, the ideas to reduce complexity in disparity search range contradict the regular computing scheme of the efficient cost aggregation methods. The second one is to provide accurate disparity estimates. This is the main challenge of the CTF methods aiming complexity reduction on disparity search range. In this study, a hierarchical CTF approach is proposed with two novel algorithmic blocks addressing the aforementioned perspectives of the complexity reduction problem.

CTF scheme aims to perform cost aggregation for only limited number of disparities for each pixel. In other words, every pixel has its own disparity candidate set to be tested and the cost aggregations are to be performed only for those candidates. On the other hand, the efficiency of the fastest cost aggregation methods comes from the reuse of the computations that are shared for the same disparity hypothesis among neighbouring pixels. Therefore, efficient cost aggregation cannot be applied to CTF strategies; since, the neighbouring pixels might not have the same disparity hypotheses. In this study, a novel recursive edge aware filtering method that is applicable to CTF methods is proposed for the efficient cost aggregation. The proposed method is motivated from the prediction of an unknown data from known data in a local neighbourhood in an efficient way so that the cost aggregation for a pixel can be performed efficiently in the absence of necessary data. The key idea of the proposed method is to share the results of cost aggregations of different disparity hypotheses. Two conditions are assumed to hold for a reliable cost aggregation with the proposed method. The first one is that the local behaviour of the matching and aggregated cost functions of the pixels allows the unknown data to be approximated or predicted in an efficient way within a small disparity range. The second assumption is that colour-wise similar pixels in a local region have similar disparity candidate sets. Based on these assumptions, the proposed filter is applied to CTF methods for efficient cost aggregation and in this way a stereo matching framework with linear time complexity is introduced.

To improve disparity estimation accuracy of the CTF scheme, the importance of disparity estimate transfer from coarse scale to fine scale is investigated. It is observed that most of the errors stem from the propagation of the erroneous estimates across the scales. Therefore, a reliability measure associated with the costs of the disparity estimations is proposed. Reliabilities are assigned to each estimation by exploiting the distribution of the costs and imposed as a soft constraint for the disparity transfer. Unlike prior art, it is proposed to use non-local methods as transferring the reliable disparity estimates from coarse scale to fine scale in order to generate candidate sets. Two approaches are proposed. First, the transfer problem is viewed as a disparity super resolution problem and the transfer is performed via state-of-the-art filtering based DSR methods. Secondly, the problem is formulated as an optimization problem in order to preserve object boundaries better and propagate proper information across the scales. Dynamic programming is utilized to solve this optimization problem efficiently. With these proposed non-local approaches, the occlusions are handled inherently. Namely, without introducing any additional computation, the proposed disparity transfer approach, which is a necessary step for CTF approaches, not only propagates reliable estimates among scales but also solves occlusion problems at the same time. This provides further decrease in the computational load.

Finally, to improve the robustness of the proposed method to various scenes, it is proposed to adapt the parameters controlling edge-preserving behaviour that are used in both cost aggregation and disparity transfer steps to provide discontinuity preserving disparity maps. The adaptation is achieved via examining the gradients at the pixel locations and determining the possible minimum gradient magnitude on an edge.

4.2 Conclusions

Disparity estimation performance and time complexity efficiency of the proposed method are analysed and compared with the state-of-the-art methods by the results of the experiments conducted on a data set composed of stereo image pairs from various scenes.

The experimental results validate the linear time complexity of the proposed method with the image size. According to the knowledge gathered from the literature so far, the proposed method is possibly the first edge-aware stereo matching method with linear time complexity. The experimental results on standard definition images show that the proposed method is the fastest approach compared to the state-of-the-art methods. The proposed method is 5 times faster than the currently fastest methods and 60 times faster than the best performing local stereo method regarding to estimation accuracy for the used data sets in the experiments. For high resolution images, these ratios increase, thanks to disparity search range independent complexity of the proposed method. Comparative experiments on high definition images show that the currently fastest methods becomes 14 times slower than the proposed method for the used data sets.

The estimation accuracy and the robustness of the proposed method are analysed by conducting tests on different data sets. It is observed that the proposed method is comparable to the state-of-the-art methods in terms of accuracy. According to experiments on Middlebury evaluation data consisting of four stereo image pairs, the proposed method has 77% more erroneous estimates relative to the top performer and 24% more relative to the second best method which has 42% more erroneous estimates relative to the top performer; this corresponds to 2.91%, 1.31% and 1.6% of the total pixels in the image, respectively. When compared to fastest methods, the relative error percentage is only 9% and this corresponds to only 0.57% of the total pixels in the image. Moreover, the experimental results on extended Middlebury data set show that the proposed method is the second best performer with only 2% of performance drop relative to top performer. This is a good indicator of the robustness of the proposed method. It should be noted that the proposed method does not have a post processing step to produce accurate disparity maps unlike the methods involved in the comparison.

Experiments on the proposed disparity transfer approaches show that the disparity estimation accuracy of the CTF based methods are improved by using proposed non-local disparity estimate transfer methods. Especially, the optimization based approach is observed to be the best approach among those with the improvement of 38% drop on the erroneous estimates over local approaches and only introducing 17% more execution time. Moreover, implicit occlusion handling behaviour of the proposed non-local methods is validated through experiments.

In summary, according to experiments, the proposed approach is a quite efficient disparity estimation method. The performance decrease compared to the gain from computation is marginal. It is indicated by the experiments that there is a considerable computation gain for the large problem sizes and the performance loss according to state-of-the-art methods is minor. The visual results of the proposed method also support that and also show that mostly the presence of fine structures causes performance drops. Therefore, as long as the scene is not composed of mostly fine structures, the estimation results of the proposed method achieves that of the state-of-the-art.

4.3 Future Directions

The two main algorithmic blocks proposed in this dissertation for the solution of the complexity reduction problem of the stereo matching can be applied to several problems.

Apart from the CTF approaches, the proposed cost aggregation method can be applied to other methods aiming complexity reduction on disparity search range. As long as the candidate disparities are available, the proposed filtering based cost aggregation can be utilized in such applications.

The proposed disparity estimate transfer method can be extended to disparity estimate transfer between stereo image pairs so that the disparity map corresponding to one image can be efficiently estimated by using the available disparity estimates of the other image. Moreover, in a similar manner, the disparity information can be propagated along frames of a stereo video so that real-time disparity estimation can be performed.

REFERENCES

[1] C. S. A. Geiger, P. Lenz and R. Urtasun. The KITTI vision benchmark suite. `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo`, 2012.

[2] B. L. Anderson and K. Nakayama. Toward a general theory of stereopsis: binocular matching, occluding contours, and fusion. *Psychological review*, 101(3):414, 1994.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.

[4] P. N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 506–512. IEEE, 1992.

[5] R. Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 11 1954.

[6] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Computer Vision, 1998. Sixth International Conference on*, pages 1073–1080, Jan 1998.

[7] M. Bleyer and M. Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *SPIE Symposium on Electronic Imaging 2005 (Videometrics VIII)*, pages 288–299, vol. 5665, 2005. Vortrag: SPIE Electronic Imaging Conference, San Jose, CA, USA; 2005-01-18 – 2005-01-20.

[8] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):128–150, 2005.

[9] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011.

[10] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, pages 1570–1577. IEEE, 2010.

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

[12] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, Nov 2001.

[13] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, Aug 2003.

[14] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[15] C. Chang, S. Chatterjee, and P. Kube. On an analysis of static occlusion in stereo vision. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 722–723, Jun 1991.

[16] C. Cigla. *Real-Time Stereo to Multi-View Video Conversion.* PhD thesis, Middle East Technical University, July 2012.

[17] C. Cigla and A. Alatan. Efficient edge-preserving stereo matching. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 696–699, Nov 2011.

[18] C. Cigla and A. A. Alatan. Multi-view dense depth map estimation. In *Proceedings of the 2Nd International Conference on Immersive Telecommunications*, IMMERSCOM '09, pages 2:1–2:6, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[19] C. Cigla, X. Zabulis, and A. Alatan. Region-based dense depth extraction from multi-view video. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V – 213–V – 216, Sept 2007.

[20] F. C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3):207–212, Jan. 1984.

[21] L. De-Maeztu, S. Mattoccia, A. Villanueva, and R. Cabeza. Linear stereo matching. In *A13th International Conference on Computer Vision (ICCV2011)*, November 6-13 2011.

[22] L. De-Maeztu, A. Villanueva, and R. Cabeza. Near real-time stereo matching using geodesic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):410–416, Feb 2012.

[23] G. Egnal and R. Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1127–1133, Aug 2002.

[24] E. Elboher and M. Werman. Cosine integral images for fast spatial and range filtering. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 89–92, Sept 2011.

[25] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–261–I–268 Vol.1, June 2004.

[26] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 858–863, Jun 1997.

[27] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. *ACM TOG*, 30(4):69:1–69:12, 2011. Proceedings of SIGGRAPH 2011.

[28] D. Geiger, B. Ladendorf, and A. L. Yuille. Occlusions and binocular stereo. In *Proceedings of the Second European Conference on Computer Vision*, ECCV '92, pages 425–433, London, UK, UK, 1992. Springer-Verlag.

[29] M. Gong and Y.-H. Yang. Real-time stereo matching using orthogonal reliability-based dynamic programming. *Trans. Img. Proc.*, 16(3):879–884, Mar. 2007.

[30] K. He, J. Sun, and X. Tang. Guided image filtering. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010.

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

[32] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599, 2009.

[33] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–74–I–81 Vol.1, June 2004.

[34] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2009. Poster presentation: IEEE

International Conference on Image Processing 2009, Cairo, Egypt; 2009-11-07 – 2009-11-10.

[35] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(2):504 – 511, 2013.

[36] W. Hu, K. Zhang, L. Sun, and S. Yang. Comparisons reducing for local stereo matching using hierarchical structure. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6, July 2013.

[37] Y.-H. Jen, E. Dunn, P. F. Georgel, and J.-M. Frahm. Adaptive scale selection for hierarchical stereo. In *BMVC*, pages 1–10, 2011.

[38] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(9):920–932, Sept. 1994.

[39] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1033–1040 vol.2, Oct 2003.

[40] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern recognition*, 19(1):41–47, 1986.

[41] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 15–18, 2006.

[42] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515 vol.2, 2001.

[43] C. Lei, J. Selzer, and Y.-H. Yang. Region-tree based stereo using dynamic programming optimization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2378–2385, 2006.

[44] S. Z. Li. *Markov Random Field Modeling in Image Analysis.* Springer Publishing Company, Incorporated, 3rd edition, 2009.

[45] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1073–1078, Aug 2004.

[46] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[47] M.-Y. Liu, O. Tuzel, and Y. Taguchi. Joint geodesic upsampling of depth images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 169–176, June 2013.

[48] J. Lu, D. Min, R. Pahwa, and M. Do. A revisit to MRF-based depth map super-resolution and enhancement. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 985–988, May 2011.

[49] J. Lu, K. Shi, D. Min, L. Lin, and M. Do. Cross-based local multipoint filtering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 430–437, June 2012.

[50] J. Lu, H. Yang, D. Min, and M. Do. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1854–1861, June 2013.

[51] A. Luo and H. Burkhardt. An intensity-based cooperative bidirectional stereo matching with simultaneous detection of discontinuities and occlusions. *Int. J. Comput. Vision*, 15(3):171–188, July 1995.

[52] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979.

[53] S. Mattoccia, S. Giardino, and A. Gambini. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. In *Proceedings of the 9th Asian Conference on Computer Vision - Volume Part II*, ACCV'09, pages 371–380, Berlin, Heidelberg, 2010. Springer-Verlag.

[54] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang. Segment-tree based cost aggregation for stereo matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 313–320, June 2013.

[55] D. Min, J. Lu, and M. Do. A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1567–1574, Nov 2011.

[56] D. Min, J. Lu, and M. N. Do. Joint histogram-based cost aggregation for stereo matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2539–2545, 2013.

[57] D. Min and K. Sohn. Cost aggregation and occlusion handling with wls in stereo matching. *Image Processing, IEEE Transactions on*, 17(8):1431–1442, Aug 2008.

[58] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Computer Vision–ECCV 2006*, pages 568–580. Springer, 2006.

[59] S. Y. Park, S. H. Lee, and N. I. Cho. Segmentation based disparity estimation using color and depth information. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, pages 3275–3278 Vol. 5, Oct 2004.

[60] C. C. Pham and J. W. Jeon. Domain transformation-based efficient cost aggregation for local stereo matching. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(7):1119–1130, July 2013.

[61] D. Sage and M. Unser. Bi-exponential edge-preserving smoother. *IEEE Transactions on Image Processing*, pages 3924–3936, 2012.

[62] D. Scharstein and R. Szeliski. Middlebury stereo vision website. `http://vision.middlebury.edu/stereo/`, 2002.

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

[64] J. A. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1998.

[65] M. Sezgin et al. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.

[66] M. Sizintsev and R. Wildes. Efficient stereo with accurate 3-D boundaries. In *Proc. BMVC*, pages 25.1–25.10, 2006. doi:10.5244/C.20.25.

[67] A. Spoerri. The early detection of motion boundaries. 1990.

[68] J. Sun, Y. Li, S. Bing, and K. H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *In CVPR*, pages 399–406, 2005.

[69] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[70] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov Random Fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068–1080, June 2008.

[71] H. Tao and H. Sawhney. Global matching criterion and color segmentation based stereo. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 246–253, 2000.

[72] P. J. Toivanen. New geodesic distance transforms for gray-scale images. *Pattern Recogn. Lett.*, 17(5):437–450, May 1996.

[73] C. Tomasi and T. Kanade. *Detection and tracking of point features.* School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[74] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[75] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–556–I–561 vol.1, June 2003.

[76] O. Veksler. Stereo correspondence by dynamic programming on a tree. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 384–390, Washington, DC, USA, 2005. IEEE Computer Society.

[77] V. M. Vergauwen, G. V. Meerbergen, M. Vergauwen, M. Pollefeys, and L. V. Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47:275–285, 2002.

[78] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[79] Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[80] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 902–909 vol. 2, June 2005.

[81] Q. Yang. A non-local cost aggregation method for stereo matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1402–1409, June 2012.

[82] Q. Yang. Recursive bilateral filtering. In *ECCV*, pages 399–413, 2012.

[83] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time O(1) bilateral filtering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 557–564. IEEE, 2009.

[84] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(3):492–504, March 2009.

[85] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister. Real-time global stereo matching using hierarchical belief propagation. In *BMVC*, volume 6, pages 989–998, 2006.

[86] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 274–279. IEEE, 1993.

[87] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):650–656, April 2006.

[88] K.-J. Yoon and I.-S. Kweon. Stereo matching with the distinctive similarity measure. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7, Oct 2007.

[89] T. Yu, R.-S. Lin, B. Super, and B. Tang. Efficient message representations for belief propagation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[90] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the Third European Conference on Computer Vision (Vol. II)*, ECCV '94, pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[91] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):1073–1079, July 2009.

[92] Y. Zhang and C. Kambhamettu. Stereo matching with segmentation-based cooperation. In *Proceedings of the 7th European Conference on Computer Vision-Part II*, ECCV '02, pages 556–571, London, UK, UK, 2002. Springer-Verlag.

[93] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *Int. J. Comput. Vision*, 75(1):49–65, Oct. 2007.