SELECTION AND FUSION OF MULTIPLE STEREO ALGORITHMS FOR
ACCURATE DISPARITY SEGMENTATION

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

BY

ARDA BİLGİN

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

NOVEMBER 2008

Approval of the thesis:

**SELECTION AND FUSION OF MULTIPLE STEREO ALGORITHMS
FOR ACCURATE DISPARITY SEGMENTATION**

submitted by **ARDA BİLGİN** 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. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering Dept., METU**

Assist. Prof. Dr. İlkey Ulusoy _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Prof. Dr. Uğur Halıcı _____
Co-Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

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

Prof. Dr. Uğur Halıcı _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. İlkey Ulusoy _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Alptekin Temizel _____
Informatics Institute Dept., METU

Uğur Güngör, MSc. in EEE _____
MGEO, ASELSAN

Date: 25.11.2008

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: Arda Bilgin

Signature:

ABSTRACT

SELECTION AND FUSION OF MULTIPLE STEREO ALGORITHMS FOR ACCURATE DISPARITY SEGMENTATION

Bilgin, Arda

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. İlkey Ulusoy

Co-Supervisor: Prof. Dr. Uğur Halıcı

November 2008, 125 pages

Fusion of multiple stereo algorithms is performed in order to obtain accurate disparity segmentation. Reliable disparity map of real-time stereo images is estimated and disparity segmentation is performed for object detection purpose. First, stereo algorithms which have high performance in real-time applications are chosen among the algorithms in the literature and three of them are implemented. Then, the results of these algorithms are fused to gain better performance in disparity estimation. In fusion process, if a pixel has the same disparity value in all algorithms, that disparity value is assigned to the pixel. Other pixels are labelled as unknown disparity. Then, unknown disparity values are estimated by a refinement procedure where neighbourhood disparity information is used. Finally, the resultant disparity map is segmented by using mean shift segmentation.

The proposed method is tested in three different stereo data sets and several real stereo pairs. The experimental results indicate an improvement for the stereo analysis performance by the usage of fusion process and refinement procedure.

Furthermore, disparity segmentation is realized successfully by using mean shift segmentation for detecting objects at different depth levels.

Keywords: Disparity Segmentation, Stereo Analysis, Disparity Map, Disparity Refinement, Mean Shift Segmentation.

ÖZ

DOĞRU DERİNLİK BÖLÜTLEMESİ İÇİN UYGUN STEREO ALGORİTMALARININ SEÇİLMESİ VE KAYNAŞTIRILMASI

Bilgin, Arda

Yüksek lisans, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. İlkay Ulusoy

Ortak Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Kasım 2008, 125 sayfa

Doğru fark bölütlemesi için uygun stereo algoritmalarının seçilmesi ve kaynaştırılması gerçekleştirilir. Gerçek zamanlı stereo görüntülerin güvenilir fark haritası kestirilir ve cisim bulma amacı için fark bölütlemesi yapılır. Önce gerçek zamanlı uygulamalarda yüksek performansa sahip stereo algoritmaları literatürdeki algoritmalar arasından seçilir ve bunların üçü uygulanır. Sonra fark kestirme işleminde daha iyi performans elde etmek için bu algoritmaların sonuçları kaynaştırılır. Kaynaştırma işleminde, eğer bir piksel tüm algoritmalarda aynı fark değerine sahipse, bu fark değeri piksele atanır. Diğer piksellerin fark değerleri bilinmeyen kabul edilir. Sonra bilinmeyen fark değerleri komşu fark bilgisini kullanan iyileştirme yordamı sayesinde kestirilir. Son olarak, sonuç fark haritası ortalama kaydırma bölütlemesi kullanılarak bölütlenir.

Önerilen yöntem 3 farklı stereo veri seti ve çeşitli gerçek stereo çiftleri için test edilir. Yapılan deneyler, kullanılan kaynaştırma işlemi ve iyileştirme yordamının stereo analiz performansını arttırdığını gösterir. Ayrıca farklı derinlikteki objeleri

bulmak için ortalama kaydırma bölütlemesi kullanılarak yapılan fark bölütlemesi başarılı bir şekilde gerçekleştirilir.

Anahtar Kelimeler: Fark Bölütleme, Stereo Analizi, Fark Haritası, Fark İyileştirme, Ortalama Kaydırma Bölütlemesi.

To My Family

ACKNOWLEDGMENTS

I express my deepest gratitude to Assist. Prof. Dr. İlkey ULUSOY for her guidance and support throughout this study.

I could never achieve this goal without support of my family. I thank them for their encouragement and full support.

I would like to thank all my friends who helped in producing this thesis for their valuable suggestions and comments.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ	vi
ACKNOWLEDGMENTS	ix
TABLE OF CONTENTS.....	x
LIST OF FIGURES	xiii
LIST OF TABLES	xvii
CHAPTERS	
1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 THESIS OVERVIEW	4
2 STEREO ANALYSIS AND IMAGE SEGMENTATION.....	5
2.1 STEREO ANALYSIS	5
2.1.1 FINDING CORRESPONDENCE	7
2.1.2 DISPARITY ESTIMATION	12
2.1.3 TRIANGULATION	13
2.1.4 COMMONLY USED STEREO MATCHING METHODS	14
2.1.5 PROBLEMS IN STEREO VISION	16
2.2 IMAGE SEGMENTATION.....	17
2.2.1 CLUSTERING METHODS	18
K-MEANS CLUSTERING	18
FUZZY C-MEANS CLUSTERING.....	19

2.2.2	REGION-BASED METHODS.....	21
	REGION GROWING	21
	SPLIT AND MERGE	23
2.2.3	GRAPH-BASED METHODS	24
	GRAPH THEORETIC CLUSTERING	25
	RECURSIVE SHORTEST SPANNING TREE (RSST).....	29
3	ALGORITHMS USED IN PROPOSED METHODOLOGY	31
3.1	STEREO ALGORITHMS.....	31
3.1.1	SAD CORRELATION-BASED STEREO VISION.....	31
	LEFT-RIGHT CONSISTENCY CHECK	32
	MULTIPLE SUPPORTING WINDOWS	34
3.1.2	DYNAMIC PROGRAMMING (DP) IN STEREO VISION	38
3.2	IMAGE SEGMENTATION ALGORITHM.....	50
3.2.1	MEAN SHIFT SEGMENTATION	50
4	FUSION OF STEREO ALGORITHMS FOR ACCURATE DISPARITY ESTIMATION AND DISPARITY SEGMENTATION	55
4.1	INTRODUCTION	55
4.2	STEREO ANALYSIS BLOCK.....	57
4.2.1	SAD CORRELATION-BASED STEREO.....	57
4.2.2	DYNAMIC PROGRAMMING.....	57
4.2.3	REGION-BASED STEREO.....	58
4.2.4	FUSION OF STEREO ALGORITHMS	59
4.2.5	IMPROVEMENT	60
4.3	DISPARITY SEGMENTATION BLOCK	60
5	EXPERIMENTS AND RESULTS	63

5.1	STEREO VISION TEST PROCEDURE	63
5.2	RESULTS OF STEREO ALGORITHMS APPLIED TO STEREO DATA SETS..	66
5.2.1	SAD CORRELATION-BASED STEREO	66
5.2.2	DYNAMIC PROGRAMMING	74
5.2.3	REGION-BASED STEREO	78
5.2.4	FUSION OF STEREO ALGORITHMS AND IMPROVEMENT	80
5.3	DISPARITY SEGMENTATION	91
5.4	STEREO ANALYSIS AND DISPARITY SEGMENTATION OF REAL-TIME STEREO PAIRS	97
5.4.1	TEST STEREO PAIR # 1	97
5.4.2	TEST STEREO PAIR # 2	100
5.4.3	TEST STEREO PAIR # 3	102
5.4.4	TEST STEREO PAIR # 4	104
5.4.5	TEST STEREO PAIR # 5	105
5.4.6	TEST STEREO PAIR # 6	107
5.5	STEREO ANALYSIS OF SOME STEREO VIDEO SEQUENCES	110
6	CONCLUSION AND FUTURE WORK	115
6.1	CONCLUSION	115
6.2	FUTURE WORK	117
7	REFERENCES	118
8	APPENDIX	124

LIST OF FIGURES

FIGURES

FIGURE 2.1 THE PARALLEL STEREO CAMERA CONFIGURATION.....	6
FIGURE 2.2 EPIPOLAR GEOMETRY.....	8
FIGURE 2.3 CONTINUITY CONSTRAINT.....	9
FIGURE 2.4 ORDERING CONSTRAINT.....	10
FIGURE 2.5 GEOMETRIC CONSTRAINT.....	11
FIGURE 2.6 DISPARITY LIMIT CONSTRAINT.....	12
FIGURE 2.7 DISPARITY ESTIMATION.....	12
FIGURE 2.8 TRIANGULATION.....	13
FIGURE 2.9 AN EXAMPLE OF REGION GROWING.....	23
FIGURE 2.10 SPLIT AND MERGE SEGMENTATION.....	24
FIGURE 2.11 A SIMPLE GRAPH WITH SIX VERTICES AND TEN EDGES.....	25
FIGURE 2.12 EFFECTS OF SCALE PARAMETER (Σ_D) ON AFFINITY MATRIX.....	27
FIGURE 2.13 THE FOUR EIGENVECTORS CORRESPONDING TO THE FOUR LARGEST VALUES OF THE AFFINITY MATRIX FOR THE DATASET IN FIGURE 2.12 (A), USING $\Sigma_D = 0.2$	28
FIGURE 2.14 THE RSST ALGORITHM.....	30
FIGURE 3.1 CORRELATION-BASED STEREO.....	32
FIGURE 3.2 LEFT RIGHT CONSISTENCY.....	33
FIGURE 3.3 CORRELATION AT OBJECT BORDER.....	34
FIGURE 3.4 TYPICAL DECISION CONFLICT AT OBJECT BORDER.....	36
FIGURE 3.5 CONFIGURATIONS WITH MULTIPLE WINDOWS.....	36

FIGURE 3.6 INITIALIZATION STEP OF DYNAMIC PROGRAMMING FOR LCS PROBLEM.....	41
FIGURE 3.7 MATRIX FILL STEP OF DYNAMIC PROGRAMMING FOR LCS PROBLEM.....	42
FIGURE 3.8 THE FINAL MATRIX AFTER FILLING STEP OF DYNAMIC PROGRAMMING FOR LCS PROBLEM.	43
FIGURE 3.9 TRACEBACK STEP OF DYNAMIC PROGRAMMING FOR LCS PROBLEM.....	44
FIGURE 3.10 ALL POSSIBLE LONGEST COMMON SUBSEQUENCES.....	45
FIGURE 3.11 A PATH REPRESENTING A MATCHING OF POINTS IN THE LEFT AND RIGHT IMAGES.	47
FIGURE 3.12 INITIALIZATION STEP OF DYNAMIC PROGRAMMING FOR ML STEREO PROBLEM.....	48
FIGURE 3.13 MATRIX FILL STEP OF DYNAMIC PROGRAMMING FOR ML STEREO PROBLEM.....	49
FIGURE 3.14 MS VECTOR AND ITS CONVERGENCE.	52
FIGURE 4.1 THE PROPOSED METHODOLOGY.....	56
FIGURE 4.2 REGION-BASED STEREO.....	58
FIGURE 4.3 AN EXAMPLE OF FUSION.	59
FIGURE 4.4 DISPARITY SEGMENTATION METHOD	61
FIGURE 5.1 DIFFERENT REGIONS OF SAWTOOTH DATA SET.....	65
FIGURE 5.2 OUTPUTS OF SAD CORRELATION-BASED ALGORITHMS FOR TSUKUBA, VENUS AND SAWTOOTH.	70
FIGURE 5.3 THE EFFECT OF WINDOW SIZE ON CORRELATION-BASED STEREO.	73
FIGURE 5.4 EFFECT OF THE CAMERA SENSOR NOISE VARIANCE ON DP FOR TSUKUBA, VENUS AND SAWTOOTH.	77

FIGURE 5.5 OUTPUTS OF REGION-BASED ALGORITHMS FOR TSUKUBA, VENUS AND SAWTOOTH.	79
FIGURE 5.6 OUTPUTS OF FUSION AND IMPROVED FUSION ALGORITHMS FOR TSUKUBA, VENUS AND SAWTOOTH.	85
FIGURE 5.7 APPLYING MEAN SHIFT SEGMENTATION TO THE WHOLE IMAGE OF GROUND TRUTH FOR TSUKUBA.....	91
FIGURE 5.8 SEGMENTATION OF GROUND TRUTH FOR TSUKUBA.....	93
FIGURE 5.9 SEGMENTATION OF GROUND TRUTH FOR VENUS.	94
FIGURE 5.10 SEGMENTATION OF GROUND TRUTH FOR SAWTOOTH.	94
FIGURE 5.11 DISPARITY SEGMENTATION FOR TSUKUBA.	95
FIGURE 5.12 DISPARITY SEGMENTATION FOR VENUS.....	95
FIGURE 5.13 DISPARITY SEGMENTATION FOR SAWTOOTH.....	96
FIGURE 5.14 TEST STEREO PAIR # 1.	98
FIGURE 5.15 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 1.	99
FIGURE 5.16 TEST STEREO PAIR # 2.	100
FIGURE 5.17 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 2.	101
FIGURE 5.18 TEST STEREO PAIR # 3.	102
FIGURE 5.19 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 3.	103
FIGURE 5.20 TEST STEREO PAIR # 4.	104
FIGURE 5.21 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 4.	105
FIGURE 5.22 TEST STEREO PAIR # 5.	106
FIGURE 5.23 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 5.	107
FIGURE 5.24 TEST STEREO PAIR # 6.	108

FIGURE 5.25 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF TEST STEREO PAIR # 6.	109
FIGURE 5.26 OUTPUTS OF SAD_MW_CON & DP ALGORITHM FOR FRAMES OF STEREO VIDEO#1.	112
FIGURE 5.27 OUTPUTS OF SAD_MW_CON & DP ALGORITHM FOR FRAMES OF STEREO VIDEO#2.	113

LIST OF TABLES

TABLES

TABLE 5.1 QUALITY MEASURES OF STEREO IMAGES.	66
TABLE 5.2 PERCENTAGE OF BAD MATCHING PIXELS OF FOUR VERSIONS OF SAD CORRELATION-BASED ALGORITHM FOR FIVE QUALITY MEASURES FOR THREE STEREO DATA SETS.	67
TABLE 5.3 THE AVERAGE RANK VALUE OF CORRELATION-BASED ALGORITHMS FOR EACH DATA SET.....	69
TABLE 5.4 THE EFFECT OF WINDOW SIZE ON CORRELATION-BASED STEREO.	71
TABLE 5.5 THE EFFECT OF THE VARIANCE OF CAMERA SENSOR NOISE ON DYNAMIC PROGRAMMING IN STEREO.....	74
TABLE 5.6 PERCENTAGE OF BAD MATCHING PIXELS OF REGION-BASED ALGORITHMS FOR FIVE QUALITY MEASURES FOR EACH STEREO DATA SET.	78
TABLE 5.7 THE FUSED ALGORITHMS AND THEIR LABELS.....	81
TABLE 5.8 PERCENTAGE OF BAD MATCHING PIXELS OF FUSION ALGORITHMS AND IMPROVED FUSION ALGORITHMS FOR FIVE QUALITY MEASURES FOR EACH DATA SET.	81
TABLE 5.9 PERCENTAGE OF BENEFIT OF IMPROVEMENT STEP FOR FIVE REGIONS FOR EACH DATA SET.	86
TABLE 5.10 PERCENTAGE OF BAD MATCHING PIXELS OF ALL ALGORITHMS ARE GIVEN ALL TOGETHER FOR FIVE QUALITY MEASURES FOR EACH DATA SET.	88
TABLE 5.11 THE AVERAGE RANK VALUE OF ALL ALGORITHMS FOR EACH DATA SET.	90
TABLE 5.12 PARAMETERS OF STEREO ANALYSIS AND DISPARITY SEGMENTATION FOR DATA SETS.	92

TABLE 5.13 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 1.....	98
TABLE 5.14 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 2.....	100
TABLE 5.15 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 3.....	102
TABLE 5.16 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 4.....	104
TABLE 5.17 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 5.....	106
TABLE 5.18 PARAMETERS OF STEREO ANALYSIS AND DISPARITY	
SEGMENTATION FOR TEST STEREO PAIR # 6.....	108

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Stereo analysis is widely used and has an important application area in computer vision such as 3D vision, reconstruction, object detection, robot navigation and control and etc. The fundamental problem in stereo analysis is to find correspondence points (stereo matching) in stereo images. Although researchers have been trying to solve this problem for many years, it has not been solved completely yet [35].

A large variety of stereo matching algorithms have been developed to cope with matching problem. These algorithms can be stated under three titles such as pixel-based, region-based and feature-based algorithms.

In pixel-based algorithms, the aim is to find the matching pixels in stereo images. The result is a dense disparity map which means that the disparities of all pixels are found. However pixel-based algorithms have generally much higher computational complexity than others. They can be divided into two classes as global and local methods [16]. Global methods use optimization techniques that apply disparity assignments on the basis of the minimization of a global cost function which is alternatively called energy function [36]. Global methods have very high computational cost. That's why they are not suitable for real time applications. On the other hand, local methods are the window-based algorithms that use the information of pixel intensity values within a finite window for determining the disparity of each pixel. Local methods realize less accurate disparity map than global

methods do, but they can run fast enough to be deployed in many real time applications [36].

Region-based algorithms, generally known as segment-based algorithms, firstly aim to separate image into regions by using segmentation techniques and then try to find the matching regions in stereo pairs. They calculate the disparities for all regions and assume that all pixels in the same region have the same disparity. Disparity map obtained by region-based algorithms is also an example for dense disparity map. In addition to these, region-based algorithms have lower computational cost than pixel-based algorithms have.

Feature-based algorithms are less used in stereo vision. They aim to extract some specific features from the reference image and then try to find matching features in other image. They construct sparse disparity map which means that the disparities are found for only some pixels in the image. They have lower computational complexity than others because of less number of points to be matched.

Stereo matching process can be facilitated by using some predefined constraints such as epipolar geometry, similarity, uniqueness, continuity, ordering, geometric and disparity limit. They are going to be explained in section 2.1.1 in detail.

The disparity map contains the depth information about the scene and it could also be useful for object segmentation because it has generally uniform information (depth information) about objects at specified depth. However the efficiency of object segmentation depends on the quality of disparity map. The less accurate disparity maps can mislead the segmentation process. Especially, in real time stereo applications for object segmentation, the most efficient and the fastest algorithms should be used.

The objective of this thesis is to develop a robust, accurate and reliable methodology which is capable of extracting the accurate disparity map of the scene for near real time applications and realizing disparity segmentation for detecting objects at different depth levels. For this purpose, stereo vision literature has been searched and analyzed as the first step. Taxonomy paper of Daniel Scharstein and

Richard Szeliski [16] is found to be the best for such an analysis since it includes comparison of various stereo algorithms. Two algorithms (SAD correlation and dynamic programming) have been selected among the algorithms in [16] by analyzing their performance in real-time stereo vision and computational complexity. These two algorithms are found to be suitable for real-time applications.

SAD correlation-based stereo has been implemented by utilizing the informations in [37]. [37] analyses the behaviour of correlation-based stereo to find ways to improve its quality while maintaining its real-time suitability. Hence, left-right consistency check and multiple supporting windows techniques are used in [37] for improving the performance of correlation-based stereo and for reducing its complexity. On the other hand, dynamic programming [21] is a global stereo method and does not have low computational cost but its performance in stereo [16] is satisfactory and it is easy to be implemented. It can also model the real world easily.

Besides SAD correlation-based stereo and dynamic programming, the region-based stereo algorithm is selected as the third stereo algorithm which is implemented in this thesis. It is very suitable for real-time application that it can enhance noise tolerance by aggregating over similar pixels and has low computational complexity [34].

In order to extract accurate disparity map of the scene, the individual disparity maps obtained by three stereo algorithms (SAD correlation-based stereo, dynamic programming and region-based stereo) are fused in this study. Afterwards, disparity map is segmented by mean-shift segmentation for detecting objects at different depths. Mean-shift segmentation algorithm is not computationally expensive and can effectively segment the color and gray-scale images. That's why mean-shift segmentation algorithm has been used in region-based stereo for choosing the initial regions and in disparity segmentation for detecting objects.

Data sets (Tsukuba, Venus, Sawtooth) used in [16], [48] are used in this thesis for determining the efficiency of implemented stereo algorithms. Different stereo pairs which were obtained from different sources are also used for determining the performance of stereo algorithms in near real time applications.

1.2 THESIS OVERVIEW

The thesis chapters are organized as follows;

Chapter 2 is devoted to literature survey for stereo analysis and image segmentation methods.

Chapter 3 includes the explanation of the methods used in this thesis.

Chapter 4 explains the proposed methodology used in this thesis.

Chapter 5 covers the results obtained from data sets and real time stereo pairs.

Chapter 6 includes the summary, conclusion, and possible future work after this study.

CHAPTER 2

LITERATURE SURVEY

STEREO ANALYSIS AND IMAGE SEGMENTATION

2.1 STEREO ANALYSIS

In human stereo vision, eyes of a human can perceive the depth of objects in the environment. Each eye captures its own view and the two separate images are sent to the brain. When the two images arrive simultaneously in the back of the brain, they are united into one picture. The mind combines the two images by matching up the similarities and adding in the small differences. The small differences between the two images add up to a big difference in the final picture. This final picture is defined as three dimensional stereo picture [8].

Human stereo vision can be imitated by using two cameras like our eyes separated exactly by a distance. Computer calculates the differences between two images and uses this information for determining depth estimation. It is called computer stereo vision or machine stereo vision. As an alternative, two or more images taken from a moving camera can also be used to compute depth information in stereo vision. In this thesis, the structure with two cameras is going to be used.

Determining the depth of various points in the scene is one of the important tasks in computer vision. Laser range finding and stereo vision are commonly used methods for determining the depth. Stereo vision has two major advantages over laser range finding. First, it is a passive method, which means that it does not alter the environment. Second, it is better for high resolution three dimensional information of moving objects because a camera can take a high resolution image in

less than a thousandth of a second, whereas the laser range finding technology usually requires objects to be relatively motionless. Therefore stereo vision is more preferable than other methods [9].

Stereo camera configuration [10] is shown in Figure 2.1. (Y_L, Z_L, X_L) is the coordinate system of left camera with its origin at O_L and (Y_R, Z_R, X_R) is the coordinate system of right camera with its origin at O_R . $P(x, y, z)$ is a point in real world and it is projected onto left and right cameras as $P_L(x_l, y_l)$ and $P_R(x_r, y_r)$ which are called corresponding points in stereo vision. Determining all corresponding points is a state of finding corresponding pairs which is explained in section 2.1.1. Displacement $(x_l - x_r)$ of P_L and P_R which is explained in section 2.1.2, can provide an estimate of the corresponding 3-D position [10]. 3-D position is actually calculated by triangulation step which is explained in section 2.1.3.

Figure 2.1 The parallel stereo camera configuration. The baseline between two cameras is b and the focal length is f [10].

There are generally three steps in stereo vision. These are;

1. Finding correspondence
2. Disparity estimation
3. Triangulation

2.1.1 FINDING CORRESPONDENCE

Matching process (finding correspondence) is the first and most significant part of stereo vision. It aims to find the corresponding point (pixel or region or feature) in the right image for each point in the left image. Approaches about correspondence problem can be broadly classified into three categories: pixel-based matching, region-based matching and feature-based matching techniques which will be explained respectively in section 2.1.4. Matching process is a very difficult search procedure so some matching constraints must be declared in order to minimize false matches. Below is a list of the commonly used constraints [13].

EPIPOLAR GEOMETRY CONSTRAINT :

Each 3-D point defines an epipolar plane intersecting each image along the epipolar line. The pixel in the right image and its corresponding pixel in the left image must lie along corresponding epipolar lines so if corresponding epipolar lines are determined, searching process for finding corresponding points gets easy because 2-D search on the whole image is transformed to 1-D search along only epipolar line. In order to determine epipolar lines, calibration of cameras has to be made and their parameters have to be known. Epipolar geometry with two cameras which are arranged arbitrarily is shown in Figure 2.2.

Figure 2.2 Epipolar Geometry [11].

When both cameras are arranged in parallel as shown in Figure 2.1, their focal points are at the same height and their focal lengths are the same, general epipolar geometry gets simple. All corresponding epipolar lines are horizontal therefore corresponding points can be found on the same scanline in the right and left images. If the cameras are not arranged in parallel, corresponding epipolar lines can fall along horizontal scan lines of the stereo images if rectification is applied [12].

SIMILARITY CONSTRAINT :

For the pixel-based approach, the matching pixels must have similar intensity values (lower than a specified threshold or minimum of cost function). For the region-based approach, the matching windows must be highly correlated. For the feature-based approach, the matching features must have similar attribute values [14].

UNIQUENESS CONSTRAINT :

Every point in the left image corresponds to a “unique” point in the right image. This means that a given point in one image can not match *more than one* point in the other image [14].

CONTINUITY CONSTRAINT :

The disparity of the corresponding points varies smoothly almost everywhere over the image. Points on object boundaries are the exceptions where the continuity constraint fails because disparity changes abruptly at boundaries. That’s why, the disparity of a matched pair and that of its nearest neighbor pair should be alike if they are not at the boundaries. For example, if (a,b) and (e,f) shown in Figure 2.3 are the nearest matching pairs on the left hand side and right hand side of the target point x, then $d(x,y)$ should not be larger than $d(a,b)$ or $d(e,f)$, where d denotes disparity, and y is the matching point for x [10].

Figure 2.3 Continuity constraint. Disparity of (x,y) should be between the disparity of matching pairs (a,b) and (e,f).

ORDERING CONSTRAINT :

The target points and their corresponding points should have the same order on the corresponding epipolar lines in both images. For example, (a,b) and (e,f), shown in Figure 2.4, are the matching pairs and if a is to the left of e then b should also be to the left of f [10].

Figure 2.4 Ordering constraint. Target point a is to the left of b so corresponding point b should be to the left of f.

The ordering constraint fails at the forbidden zone in the field of view of the camera which means that if one object exists much closer to the camera than other objects, ordering constraint shall fail [13].

GEOMETRIC CONSTRAINT :

Geometric constraint requires that the corresponding point of a pixel in the right image should be searched to the left of the same pixel location in the left image. This means that the x-coordinate of corresponding points in the right image must be smaller than x-coordinate of target points in the left image and vice versa. Figure 2.5 illustrates geometric constraint [10].

Figure 2.5 Geometric constraint. The column index of the corresponding point $a'(y, x_r)$ in the right image must be smaller than the column index of the target point $a(y, x_l)$ in the left image.

DISPARITY LIMIT CONSTRAINT :

The search space may be reduced significantly by limiting the disparity range in order to find corresponding points. This constraint is related with the distance between the left and right cameras. For example, in Figure 2.6, $a(y, x_l)$ is a target point in the left image. Corresponding point $a'(y, x_r)$ should be looked for in the range $x_l \leq x_r \leq x_l - \text{max_disparity}$ along the corresponding epipolar line in the right image. Here “max_disparity” is the limit of the disparity range.

Disparity limit constraint reduces the computational complexity of the matching algorithms and the likelihood of false matches [13].

Figure 2.6 Disparity limit constraint.

2.1.2 DISPARITY ESTIMATION

By using the matching algorithms in stereo vision, anyone can extract the disparity map of scene which is the basic clue for depth. Disparity means displacement between two corresponding points in the left and right images. For example, Figure 2.7 shows corresponding scanlines extracted from a pair of rectified stereo images. $A(y_l, x_l)$ and $B(y_r, x_r)$ are the corresponding points and $y_l = y_r$ because of epipolar constraint. Disparity d between A and B is equal to the difference of x -coordinate indexes x_l and x_r .

Figure 2.7 Disparity estimation.

Disparity map is the collection of estimated disparities in the scene. There are two categories for extracting disparity maps. These categories are dense stereo algorithms and sparse stereo algorithms. While dense stereo algorithms calculate the disparities of all pixels in the scene, sparse stereo algorithms calculate the disparities of some pixels. Pixel-based matching and region-based matching techniques, as shown in section 2.1.4, are the examples of dense stereo however feature-based matching technique is an example of sparse stereo.

2.1.3 TRIANGULATION

The technique which can obtain depth information by using two stereo images is called triangulation [15]. In Figure 2.8, the scene point $P(x,y,z)$ in the real world is observed at points $p_l(x_l, y_l)$ and $p_r(x_r, y_r)$ in the left and right stereo image planes. Let us assume that the origin of the coordinate system coincides with the left camera center C_l , the distance between the left and right cameras is b and focal length of these cameras is f . The depth information (z) of P can be calculated by using the focal lengths and positions of the two cameras.

Figure 2.8 Triangulation [15].

Comparing the similar triangles C_lMp_l - PKp_l and NC_rP_r - KPP_r

$$\frac{f}{x_l} = \frac{z-f}{x-x_l} \quad \Rightarrow \quad \frac{x}{z} = \frac{x_l}{f} \quad (2.1)$$

$$\frac{x_r}{f} = \frac{x-b-x_r}{z-f} \quad \Rightarrow \quad \frac{x-b}{z} = \frac{x_r}{f} \quad (2.2)$$

Combining Eq. (2.1) and Eq. (2.2), depth of point P can be computed as follows;

$$z = \frac{fb}{x_l - x_r} \quad (2.3)$$

Here, $x_l - x_r$ is the disparity. According to Eq. (2.3), we understand that depth information is inversely proportional to disparity and that's why disparity is an important clue for depth.

2.1.4 COMMONLY USED STEREO MATCHING METHODS

Stereo matching has long been one of the central research problems in computer vision. Early works were motivated by the desire to recover depth maps and shape models for robotics and object recognition applications. Nowadays, stereo matching algorithms have been developed for many stereoscopic applications such as 3D vision, teleconferencing, segmentation and image-based rendering.

There are three basic stereo matching methods in the literature which are pixel-based, region-based and feature-based matching methods. They are separated from each other according to matching different points like pixels, regions and features.

In the pixel-based stereo methods, two types of stereo algorithms exist. These are local (window-based) and global methods [16] which perform the following four steps.

1. Matching cost computation,
2. Cost aggregation,
3. Disparity computation / Optimization,
4. Disparity refinement.

Local methods [17] usually utilize intensity values of pixels for computing the matching cost function by using the traditional sum of squared differences (SSD), sum of absolute differences, etc. These methods also aggregate the information of multiple pixels using smoothness constraints. The simplest cost aggregation aims to minimize the matching error within fixed size rectangular windows [18]. Better approaches about local methods utilize multiple windows [19], adaptive windows [20] which change their size in order to minimize the error and have high performance at discontinuities. In local methods, disparities are computed by selecting the minimal (winner take all (WTA) optimization) aggregated value at each pixel.

Global methods realizes matching cost computation for each pixel. They try to find disparities of pixels that minimizes a cost function (usually selected as energy fuction) by realizing some optimization techniques. They do not use cost aggregation. Depending on the form of the energy function, the most efficient energy minimization scheme can be chosen. Some mostly used optimization techniques are dynamic programming [21], belief propogation [22], maximum flow [23] and graph cuts [24].

After obtaining the disparity map of the scene, the ways of post-processing of the computed disparities can be realized by disparity refinement step. For instance, occluded areas can be detected using left-right consistency check method, a median filter can be applied to remove spurious mismatches and occluded regions can be filled by surface fitting or by distributing neighboring disparity estimates [16].

In the region-based stereo methods, segmentation techniques, defined in section 2.2, are commonly used for obtaining regions in images. Aim is to find the corresponding regions in the left and the right images. Some techniques propose considering the regions resulting from segmentation of a reference image (left or right) and finding their correspondences without segmenting the other image [6]. On the other hand, some of region-based methods segment both images and then find corresponding regions in both images [7]. Traditionally, the disparity information of each pixel in the common region is supposed to be same. Region-based techniques have three important effects. First it reduces the ambiguity in textureless regions, secondly it reduces the computational complexity because of big regions and finally noise tolerance is enhanced by aggregating over similar colored pixels [34]. Generally correlation cost functions are used as a similarity function in matching process of regions.

In the feature-based stereo methods, distinctive features are firstly extracted from the reference image by applying some feature detector algorithms then matching process is applied for finding corresponding feature in the other image. Only the pixels of features are observed to calculate disparity. Some examples for features may be edge elements, corners, line segments etc. Canny edge detector [1], the most popular one, Sobel edge detector and others like [2] are observed as edge detection algorithms in the literature. [3] and [4] are observed as corner detection algorithms and [5] are observed as a line detection algorithm. These are mostly used feature detectors in the stereo vision field.

Dense disparity maps have the disparity information for all of the pixels but sparse disparity maps don't. So pixel-based and region-based methods are examples of dense disparity maps and feature-based method is an example of sparse disparity maps. In this thesis, only the algorithms of dense disparity maps will be used.

2.1.5 PROBLEMS IN STEREO VISION

There are many problems in matching process. Some of them are listed below [47]:

Occlusions : For stereo vision, an occlusion is defined as a point in the reference image which has no corresponding point in the other image. Occlusions cause several false matchings in matching algorithms. Handling occlusions is a hard problem in stereo vision.

Brightness variations : In general, cameras often have different orientations relative to the source of illumination [47] so a point in the reference image may have different color value from its corresponding point in the other image. Therefore brightness variations make matching process difficult.

Perspective effects : There exist distortions due to perspective effects of the cameras [47].

Ambiguities : Repetitive patterns in the scene cause ambiguity in matching process [47]. Most existing algorithms are still not robust enough to handle this problem.

2.2 IMAGE SEGMENTATION

Image segmentation can be considered as the first step and the key issue in object recognition, scene understanding and image understanding. The aim of image segmentation can be defined as partitioning an image into homogeneous and meaningful regions in terms of the features of the pixels. The pixels in a region are similar to each other with respect to some characteristic or computed property, such as color, intensity, or texture.

Image segmentation is one of the most challenging problems of image processing and many different approaches and methods have been proposed in the literature for handling this problem. However there is still not an exact solution which can be applied to all image types and obtains perfect results.

Generally image segmentation methods can be divided into three categories:

1. Clustering methods,
2. Region-based methods,
3. Graph-based methods.

2.2.1 CLUSTERING METHODS

Clustering-based image segmentation partitions a set of pixels (having similar values) into groups. The components of these pixels may be intensity values, RGB values and color properties, some other computed properties and texture measures or a combination of these components. The similarity criteria [25] of the pixels are generally determined by the absolute or squared difference of pixels' components. There are two types of clustering in the literature. These are hierarchical and partitional techniques.

In hierarchical clustering [26], [27], the number of clusters is not specified at the beginning of the clustering process. It builds (agglomerative), or breaks up (divisive), a hierarchy of clusters. The traditional representation of this hierarchy is a tree (called a dendrogram), with individual elements at one end and a single cluster containing every element at the other. Agglomerative algorithms begin at the top of the tree (each pixel is supposed to be an individual cluster at the beginning), whereas divisive algorithms begin at the root (all pixels are supposed to be elements of one cluster at the beginning). Dendrogram is cut at a specified level (selected precision) at the end of the hierarchical algorithm for obtaining clusters.

In partitional clustering, the number of clusters is specified at the beginning of the clustering process. Partitional algorithms generally try to insert pixels to a cluster by observing the minimization of a specified function. There are two mostly used partitional clustering algorithms (k-means clustering and fuzzy c-means clustering) which will be explained below.

K-MEANS CLUSTERING

The k-means algorithm assigns each pixel in the image to the cluster whose center (also called centroid) is nearest [29],[30]. The center of clusters is the average of all of the pixels in the cluster, that is, its coordinates are the arithmetic mean for

each dimension separately over all of the pixels in the cluster. The steps of k-means clustering algorithm are listed below [26]:

1. Select the clusters number K and the initial cluster centers either randomly or based on some heuristic.
2. Assign each pixel in the image to the cluster that minimizes the variance between the pixel and the cluster center. Variance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors.
3. Compute the new cluster centers.
4. Repeat steps 2 and 3 until the algorithm has converged. Convergence is occurred if none of the cluster centers changes anymore.

Convergence may not always exist in the k-means clustering algorithms so a maximum number of iterations can be assigned for stopping without convergence [28].

FUZZY C-MEANS CLUSTERING

In k-means clustering, all pixels have two states for belonging to a cluster. However in fuzzy c-means clustering [28], each pixel has a degree of belonging to clusters (as in fuzzy logic) rather than belonging completely to just one cluster. Pixels have a coefficient $u_k(x)$ giving the degree of being in the kth cluster. For observed pixel, the sum of its coefficient degrees is equal to 1 which is shown in Eq. (2.4).

$$\sum_{k=1}^c u_k(x_i) = 1 \quad \begin{array}{l} i = 1, \dots, n \\ n = \text{total no. of pixels} \\ c = \text{total no. of clusters} \end{array} \quad (2.4)$$

The centroid of a cluster (y_k) is determined by calculating the mean of all pixels' coefficient degrees belonging to that cluster, shown in Eq. (2.5). The degree of belonging is related to the inverse of the distance to the cluster, as given in Eq. (2.6). Distance measure d is the Euclidian distance.

$$y_k = \frac{\sum_{i=1}^n u_k(x_i)^m x_i}{\sum_{i=1}^n u_k(x_i)^m} \quad \begin{array}{l} m \in (1, \infty) \\ m = \text{control parameter of fuzziness} \end{array} \quad (2.5)$$

$$u_k(x_i) = \frac{1}{d(y_k, x_i)} \quad (2.6)$$

After the coefficient degrees are updated, they must be normalized and fuzzyfied with parameter m therefore their sum is equal to 1. Normalized coefficient formula is shown in Eq. (2.7)

$$u_k(x_i) = \frac{1}{\sum_{j=1}^c \left(\frac{d(y_k, x_i)}{d(y_j, x_i)} \right)^{2/(m-1)}} \quad (2.7)$$

The steps of fuzzy c-means clustering algorithm are listed below [28]:

1. Select the number of clusters and assign randomly to each pixel coefficients for being in the clusters.
2. Determine the centroid of each cluster by using Eq. (2.5).
3. For each pixel, determine its coefficients of being in the clusters by using Eq. (2.6) and Eq. (2.7).
4. Repeat steps 2 and 3 until the algorithm has converged. Convergence is occurred if the coefficients' change between two iterations is no more than a specified sensitivity threshold.

In hierarchical clustering [26], once a pixel is assigned to a cluster incorrectly at the beginning of the clustering, there is no way to correct the error. This is the disadvantage of hierarchical clustering however in partitional clustering, incorrect assignments of pixels can be corrected by optimizing some criterion functions. Besides this advantage of partitional clustering, the determination of the number of clusters and also the centroid of clusters are disadvantages. The number and centroids of clusters may be determined by histogram-based image techniques but these are not always reliable.

Clustering methods like k-means and fuzzy c-means, do not use spatial information so the resulting segments might not be connected and can be widely scattered. In addition to these, clustering methods can be sensitive to noise and intensity inhomogeneities [28].

Mean-shift segmentation, a type of clustering method, is a nonparametric technique and uses spatial information of pixels so it often gives a more satisfactory segmentation result. MS segmentation technique is used in this study and will be explained in Chapter 3 in detail.

2.2.2 REGION-BASED METHODS

Region-based methods attempt to group pixels into homogeneous regions such that these pixels share some similar properties. Region growing and split and merge algorithms are the most commonly used types of region-based segmentation methods.

REGION GROWING

Region growing [31] is one of the most simple and popular region-based segmentation algorithms. It starts by choosing a (or some) starting point (or seed pixel) and aims to fill the homogeneous region by adding neighbor pixels according to some similarity measure $S(i,j)$.

First, consider a pixel r adjacent to pixel q . Pixel r can be added to pixel q 's region iff $S(q, r) > T$ or $S(q, r) < T$ for some threshold T . It can then proceed to the other neighbors of q .

Here are some problems in region growing technique [31]. How can the similarity measure and threshold value be defined? While adding q 's neighbor pixel r to the region, should the similarity of r with the seed or the similarity of r with q be used?

For the first question, similarity measure can be generally selected as the difference of pixel intensities, RGB values or texture properties. Threshold value can be modified according to the properties of image or application [31].

For the second question, one approach is to always compare back to the seed point p by using $S(p, r)$ when adding pixel r to the growing region so the choice of seed pixel becomes very important for gaining satisfactory results from region growing. Other approach is to always compare pixel r to the neighbor pixel q by using $S(q, r)$ so region can grow farther away from the original seed pixel. Some other approaches compare the pixel r with the average of pixels in the region. This is an alternative technique in region growing [31].

The selection of seed pixels is also very important in region growing. Cluster methods can be used for selection of seed pixels because the center of clusters is ideal locations for seed pixels.

There is an example of region growing in Figure 2.9. The positions of seed pixels are selected as top left corner and the absolute difference between the seed pixel and adding pixel should be less than T in Figure 2.9.

Figure 2.9 An example of region growing. (a) Original image array, (b) Segmentation result for threshold $T = 3$, (c) Segmentation result for threshold $T = 8$.

SPLIT AND MERGE

Image can be split into regions by controlling the predicate P which measures the region homogeneity. If this predicate is not satisfied for some region, it means that that region is inhomogeneous and should be split into subregions (four quadrants). On the other hand, if the predicate is satisfied for the union of two adjacent regions, then these regions are collectively homogeneous and should be merged into a single region. This method is called split and merge [32] which is shown in Figure 2.10.

Figure 2.10 Split and merge segmentation. (a) Original image, (b) Initial split in four squared blocks, (c) Splitting of the image in homogeneous blocks, (d) Final segmentation after the merging [32].

2.2.3 GRAPH-BASED METHODS

A graph is a collection of vertices (nodes) and a collection of edges (links) which connect pairs of vertices. It can be represented as $G = (V, E)$, where V is the vertices and E is the edges between vertices as shown in Figure 2.11.

Figure 2.11 A simple graph with six vertices and ten edges.

There are so many graph types in the literature, some of them (*weighted, directed, undirected, complete and partial*) will be explained here. In a *weighted graph*, the edges are weighted and the connection strength among vertices is modeled via corresponding weights. If the direction of edges is specified by arrows, then the graph is denoted as *directed graph*, if the direction is not specified, then the graph is called *undirected graph*. In a *complete graph*, all vertices are connected each other however in a *partial graph*, each vertex is not necessarily linked to every other vertex.

In order to analyze image in graph theory, image is mapped onto a graph. Each pixel in the image corresponds to a vertex and weights of edges correspond to similarity measure between the pixels. There are a lot of graph-based algorithms in the literature, two mostly used algorithms (graph-theoretic clustering [29] and recursive shortest spanning tree [33]) will be explained in this section.

GRAPH THEORETIC CLUSTERING

In graph theoretic clustering [29], the undirected weighted graph is used for image analysis and the graph can be represented by a square matrix (affinity matrix)

that row and column represents each vertex. The $(i,j)^{th}$ element of the matrix represents the weight on the edge from vertex i to vertex j .

When segmentation is used as simple clustering, it is needed to supply some measure of how similar clusters were. So, the weights of edges which are called affinity measures, should be determined. The weight of an edge connecting similar vertices should be large, and the weight of an edge connecting very different nodes should be small. Therefore there are some affinity functions that generates affinity measures according to some properties of pixels. These are shown below [29];

$$\begin{aligned}
\text{Affinity by intensity} & : \quad aff(x,y) = \exp \left\{ - \left(\frac{1}{2 \sigma_i^2} \right) (|| I(x) - I(y) ||^2) \right\} \\
\text{Affinity by color} & : \quad aff(x,y) = \exp \left\{ - \left(\frac{1}{2 \sigma_c^2} \right) (|| c(x) - c(y) ||^2) \right\} \\
\text{Affinity by distance} & : \quad aff(x,y) = \exp \left\{ - \left(\frac{1}{2 \sigma_d^2} \right) (|| x - y ||^2) \right\} \\
\text{Affinity by texture} & : \quad aff(x,y) = \exp \left\{ - \left(\frac{1}{2 \sigma_t^2} \right) (|| t(x) - t(y) ||^2) \right\}
\end{aligned}$$

σ is a scale parameter which will be large if quite distant points should be grouped and small if only very nearby points should be grouped.

Different values of σ affects the affinity matrix as shown in Figure 2.12. In Figure 2.12, large values of affinity matrix are demonstrated as light and small values are demonstrated as dark. Closest light blocks on the diagonal represents the clusters. Four clusters of 10 points are shown in the dataset (Figure 2.12 (a)), and these clusters are obviously seen when $\sigma_d = 0.2$ in Figure 2.12 (c).

Figure 2.12 Effects of scale parameter (σ_d) on affinity matrix. (a) Data sets consists four groups of ten points each, (b) $\sigma_d = 0.1$, (c) $\sigma_d = 0.2$, (d) $\sigma_d = 1$ [29].

After examining some mathematical formulation about affinity matrix [29], it can be seen that a cluster can be formed by obtaining the eigenvector with the largest eigenvalue of affinity matrix. So, if the number of clusters is known, clusters can be extracted from the eigenvectors of affinity matrix. The elements of eigenvector which belongs to observed cluster, are far away from zero and others which do not belong to observed cluster, are nearly zero. This is shown in Figure 2.13 for dataset in Figure 2.12 (a).

Figure 2.13 The four eigenvectors corresponding to the four largest values of the affinity matrix for the dataset in Figure 2.12 (a), using $\sigma_d = 0.2$ [29].

If the largest eigenvalues of the affinity matrix are very similar, eigenvectors can not be used for specifying the clusters' elements. In this case, *normalized cuts* which is explained in [29], is an effective method for determining clusters.

RECURSIVE SHORTEST SPANNING TREE (RSST)

Image is mapped onto the undirected weighted graph with number of vertices equal to the number of pixels in the image and four way connected graph is used which means that a vertex is connected to at most four vertices which are left, right, top and bottom vertices.

The steps of RSST [33] is listed below:

1. The edge weights between the vertices are calculated as the absolute values of the grey level differences between vertices. Vertices weights are intensity values of pixels.
2. Edges are sorted according to their weights and the weakest edge with the smallest weight value is detected.
3. The vertices connected to each other through weakest link are merged as a region and the vertex weight of the merged region is updated as the mean intensity of the vertices.
4. Edge weights are updated and duplicated edges are removed.
5. Repeat steps 2,3 and 4 until the desired number of vertices which is the required number of regions, is obtained.

The RSST algorithm is shown in Figure 2.14.

Figure 2.14 The RSST algorithm.

CHAPTER 3

ALGORITHMS USED IN PROPOSED METHODOLOGY

3.1 STEREO ALGORITHMS

3.1.1 SAD CORRELATION-BASED STEREO VISION

Correlation-based algorithms match image windows centered at an observing pixel and realize dense disparity map of the scene. They compare a window in the reference image to a shifting window in the other image according to a minimization of correlation cost function. The shifting window is moved in the other image along the corresponding epipolar line by an incrementing integer which is called a candidate of disparity values. The calculated disparity is known to minimize the correlation cost function and to be the displacement of observing pixel which is at the center of correlation window.

In correlation-based algorithms, there are various similarity measures which are also known as cost functions. In this thesis, sum of absolute difference (SAD) (a well-known one) cost function which produces reliable results in a minimum computation time is used.

Assume that $I_1(x,y)$ and $I_2(x,y)$ denote the intensity values of pixel at a coordinate (x,y) in the left and right image respectively as shown in Figure 3.1. The correlation window has dimensions $(2n+1)*(2m+1)$. Therefore the indexes which appear in Eq. (3.1) vary between $-n$ and $+n$ for the i -index and $-m$ and $+m$ for the j -index.

Figure 3.1 Correlation-based stereo.

$$SAD : C(x,y,d) = \sum_{i,j} [I_1(x+i,y+j) - I_2(x-d+i,y+j)] \quad \text{where } d=\text{disparity} \quad (3.1)$$

The disparity value of pixel at (x,y) in the left image minimises the SAD cost function which is shown in Eq. (3.2).

$$d_c(x,y) = \min_{(d)} (C(x,y,d)) \quad (3.2)$$

Correlation-based algorithms can not solve the problem of occlusion and depth discontinuities at the boundaries of objects. The location of object borders is important for object recognition and segmentation purposes so [37] solves this problem by improving correlation-based stereo. Improving process uses two most important techniques which are left-right consistency check and multiple supporting windows.

LEFT-RIGHT CONSISTENCY CHECK

Occlusions create points that do not belong any conjugate pairs [19]. This situation will result false matches in matching process. Consistency check is a well known method for determining occlusion regions in the image and widely used in

stereo vision. Occlusions occur at boundaries of objects so determining occlusions helps to specify and recognize the objects.

This method is explained particularly in [19]. Generally the points in the reference image (stated as left image in this thesis) are observed and matched with corresponding points in the other image (stated as right image in this thesis). If the role of left and right images is reversed, the points in the right image must have the same disparity with the corresponding points in the left image. It is worthwhile noting that the latter is equivalent to the uniqueness constraint, which states that each point on one image can match at most one point on the other image.

For example, a point K, shown in Figure 3.2, is an occluded point in the left image. Although it has no corresponding point in the right image, the minimization of cost function procedure (stated as SAD procedure in this thesis) matches it to some false point (L') anyhow. If left-right consistency check is used, one can see that correspondence point of L' is different (L) in the left image by searching from right to left [19].

Figure 3.2 Left Right Consistency. Point K which is occluded, is given L' as a match, but L' matches $L \neq K$.

For each point on the left image the disparity $d_l(x)$ is computed by using correlation algorithm. The process is repeated after reversing the two images. If $d_l(x) = d_r(x - d_l(x))$ holds, the point keeps its computed left disparity, otherwise it is marked as occluded [19].

MULTIPLE SUPPORTING WINDOWS

Fixed window correlation algorithms have a poor performance at depth discontinuities. An ideal size of window should be large enough in textureless regions but be small across the depth boundary so the question is how to handle these two problems by using window correlation. Several window-based stereo algorithms exist in the literature for answering this question, some of them uses adaptive window like [20] and some of them uses variable window like [38]. Also there is a good solution in [37] and will be explained shortly in this thesis.

Firstly the problem of correlation at depth discontinuity will be examined.

Figure 3.3 Correlation at object border [37].

In Figure 3.3, pixel of interest is on the object in the left image and correlation window is correlated along the epipolar line in the right image for determining correspondence. The position where the correlation cost function is minimum specifies the corresponding pixel. Here observing pixel (pixel of interest) is close to the object border and the left part is different in both windows because left part of correlation window in the left image is occluded in the right image. Therefore this situation causes some errors in calculation of the cost function and false matching occurs. The usage of small correlation window reduces the problem at object borders but not in whole image (especially in areas with constant depth) [37].

Another point of view is shown in Figure 3.4. Here, pixel of interest is outside the object and closer to boundary. The correct corresponding position for the correlation window R would be L but it sometimes has a mismatch with correlation window L'. Window is splitted into two regions to understand the problem easily. Let (c,a) be the correlation value of the area a and b, where a low value corresponds to a high similarity. The values $c(R_1, L_1)$ and $c(R_2, L_2')$ should be very low, because the corresponding regions are correctly matched. The matching decision of R between the window L and L' are related with the similarity of R_2 and L_2 and the similarity of R_1 and L_1' . The areas L_2 and L_1' are occluded in the right image. If $c(R_2, L_2)$ is higher than $c(R_1, L_1')$, then the wrong position L' will be chosen. Image noise will affect the decision, but it depends mostly on the similarity between the occluded background, the visible background and the object [37].

Figure 3.4 Typical decision conflict at object border [37].

Parts of the correlation window which overlap with depth discontinuity regions have errors in correlation calculation. The error can be reduced by not adding these overlapping parts into calculation. This solution is done systematically and comparably in [37] by using multiple windows technique.

Figure 3.5 Configurations with multiple windows [37].

Figure 3.5 b. shows a configuration with one small window in the middle (C_0), surrounded by four partly overlapping windows (C_{li}). The correlation value C can be computed by adding the values of the two best surrounding correlation windows (whose correlation functions are smaller than others) C_{li1} and C_{li2} to the middle one which is shown in Eq. (3.3). This approach can also be seen as using a small window C_0 and supporting the correlation decision by four nearby windows [37].

$$C = C_0 + C_{li1} + C_{li2} \quad (3.3)$$

Another configuration using nine supporting windows is shown in Figure 3.5 c. The correlation value is in this case calculated by adding the four best surrounding correlation values to the middle one which is shown in Eq (3.4) [37].

$$C = C_0 + C_{li1} + C_{li2} + C_{li3} + C_{li4} \quad (3.4)$$

The approach can be extended by adding another ring of surrounding windows as shown in Figure 3.5 d. The eight best values of the outer ring are additionally added which is shown in Eq. (3.5) [37].

$$C = C_0 + C_{li1} + \dots + C_{li4} + C_{2k1} + \dots + C_{2k8} \quad (3.5)$$

The middle window is used for reducing the blurring effect but has errors at depth discontinuity, on the other hand best surrounding windows reduce errors in depth discontinuity but have no support for decreasing blurring. According to information in [37], it is understood that the configuration of using 5 windows is suitable for real time implementation because of its computational complexity.

In Chapter 5, the implementation of correlation-based algorithm with using left-right consistency check and the multiple window configuration (5 windows technique) will be shown and results will be presented in the same chapter.

3.1.2 DYNAMIC PROGRAMMING (DP) IN STEREO VISION

Dynamic programming is the most popular technique for solving optimization problems in mathematics and computer science literature. It solves the problems which have two states named overlapping subproblems and optimal substructure. Optimal substructure means that the problem is firstly divided into subproblems and optimal ones are selected and used for the purpose of arriving the result of the whole problem. Also optimal subproblems are used together in the way of solution which means overlapping subproblems [39].

For instance; fibonacci sequence [39] is a sequence of numbers which are calculated by summing previous two numbers of the sequence itself. The value of $Fib(0)$ and $Fib(1)$ are equal to 1. $Fib(2) = Fib(1) + Fib(0)$, $Fib(3) = Fib(2) + Fib(1)$, $Fib(4) = Fib(3) + Fib(2)$ and so on. For calculating $Fib(2)$, the information of $Fib(1)$ and $Fib(0)$ are required which are optimal substructures and these are also used together in calculating $Fib(2)$ which is an example of overlapping subproblems.

$Fib(2)$ is used to calculate $Fib(3)$ and $Fib(4)$ and recalculated for each process. This may waste time insufficiently. In order to avoid this situation, an optimization technique called memoization [40] is used for speeding up the solution. It caches the results of overlapping subproblems in order to retrieve and reuse.

Fibonacci sequence, shortest path in graph theory, longest common subsequence and longest increasing subsequence in string algorithms, matrix-chain multiplication, RNA-DNA prediction algorithms are some of the problems solved by dynamic programming in the literature. To understand the principle of dynamic programming and working parts of its process easily, longest common subsequence (LCS) problem may be a good example which is shown below.

A sequence $P = \langle p_1, \dots, p_k \rangle$ is a subsequence of another sequence $A = \langle a_1, \dots, a_m \rangle$ if it is obtained by deleting some elements of A , but not changing the order of elements. The sequence $P = \langle p_1, \dots, p_k \rangle$ is a common subsequence of $A = \langle a_1, \dots, a_m \rangle$ and $B = \langle b_1, \dots, b_n \rangle$ if P is a subsequence of both A and B . To find the longest common subsequence of P , Theorem 1 is adopted [41].

Theorem 1.

$A = \langle a_1, \dots, a_m \rangle$

$B = \langle b_1, \dots, b_n \rangle$

$P = \langle p_1, \dots, p_k \rangle$

Let A , B and P be sequences as given above. If P_l is an LCS of A_i and B_j (where $i \leq m, j \leq n, l \leq k$), then

1. If $a_i = b_j$ then $p_l = a_i = b_j$, and P_{l-1} is an LCS of A_{i-1} and B_{j-1}
2. If $a_i \neq b_j$ then
 - a. $p_l \neq a_i$ implies P_l is an LCS of A_{i-1} and B_j
 - b. $p_l \neq b_j$ implies P_l is an LCS of A_i and B_{j-1}

Let $c[i,j]$ is the length of LCS of A_i and B_j . According to Theorem 1. [41];

$$c[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ 1 + c[i-1,j-1] & \text{if } a_i = b_j \text{ and } i,j > 0 \\ \max(c[i-1,j], c[i,j-1]) & \text{if } a_i \neq b_j \text{ and } i,j > 0 \end{cases}$$

This is a solution of LCS problem and it is also converted to a recursive algorithm whose pseudocode is given below;

$LCS(A_m, B_n)$

begin {1}

if ($m \neq 0$) *and* ($n \neq 0$) *then* {2}

```

    if  $a_m = b_n$  then                                {3}
         $c[m,n] = LCS(A_{m-1}, B_{n-1}) + 1$           {4}
    else                                                {5}
         $c[m,n] = \max(LCS(A_{m-1}, B_n), LCS(A_m, B_{n-1}))$  {6}
    end if                                             {7}
else                                                {8}
     $LCS(A_m, B_n) = 0$                                 {9}
end if                                             {10}
end                                                {11}

```

In LCS problem, $c[i,j]$ is assumed to be the cost function and it has a collection of feasible solutions that optimizes (minimizes or maximizes cost function) the previous cost functions. This situation is an example of optimal substructure part in dynamic programming. According to recursive algorithm shown in pseudocode, it can be seen in lines 4 and 6 that previous subproblems are used for solving current problem. This situation is also an example of overlapping subproblems part in dynamic programming. Therefore it is understood that LCS problem can be easily solved by dynamic programming.

The following is an example of longest common subsequence. There are two sequences A and B whose lengths are 7 and 6.

A = KLMLNKL

B = LNMKLL

Dynamic programming has three steps, initialization, matrix fill and traceback. Here, these steps will be analyzed and understood by solving the problem of LCS for A and B [42].

INITIALIZATION :

This is the first step in the dynamic programming approach of LCS which creates a matrix with $\text{length}(A) + 1$ columns and $\text{length}(B) + 1$ rows. The 0th row and

column of the matrix can be initially filled with 0 because of the condition, $c[i,j] = 0$ if $i=0$ or $j=0$ in the LCS theorem [42].

	0	1	2	3	4	5	6	7
		K	L	M	L	N	K	L
0	0	0	0	0	0	0	0	0
1 L	0							
2 N	0							
3 M	0							
4 K	0							
5 L	0							
6 K	0							

Figure 3.6 Initialization step of dynamic programming for LCS problem.

MATRIX FILL :

Matrix fill step finds the cost value of each position in the matrix according to cost function $c[i,j]$ of LCS problem. Each position in the matrix can be specified as a subproblem of LCS. This step starts in the upper left hand corner in the matrix and finds cost values of subproblems. In this way, matrix can be filled properly [42].

Figure 3.7 Matrix fill step of dynamic programming for LCS problem. (a) Matrix fill step of 2. row and column, (b) Matrix fill step of 3. row and column.

In Figure 3.7, using the information of cost function $c[i,j]$, the score at position (1,1) can be calculated. Since the first row (L) and first column (K) elements are different, cost value of position (1,1) must be equal to maximum of positions (0,1) and (1,0) ($\max(c[0,1], c[1,0])$) which are equal to 0. So, one of them may be a choice for position (1,1). That's why the vertical and horizontal arrows which specify the up and left positions ((0,1) and (1,0)), must be set for position (1,1). The same situation is valid for positions (2,1), (3,1), (4,2), (2,3) and (2,4).

4th row and 1st column have the same element (K). So, the score at position (4,1) must be equal to $c[3,0] + 1$ and diagonal arrow must be set for specifying the position (3,0). The same situation is valid for positions (6,1), (1,2), (1,4), (1,7) and (2,5).

5th row (L) and 1st column (K) have different elements so the score at position (5,1) must be equal to $\max(c[4,1], c[5,0])$. The up arrow must be set because (4,1) is greater than (5,0). The same situation is valid for (6,2), (2,2) and, (3,2).

1st row (L) and 3rd column (M) have different elements so the score at position (1,3) must be equal to $\max(c[1,2], c[0,3])$. The left arrow must be set because (1,2) is greater than (0,3).

The same situation is valid for (1,5), (1,6), (2,6) and (2,7).

		0	1	2	3	4	5	6	7
			K	L	M	L	N	K	L
0		0	0	0	0	0	0	0	0
1	L	0 ← 0	↑ 0	1 ← 1	↑ 1	1 ← 1	↑ 1	1 ← 1	↑ 1
2	N	0 ← 0	↑ 0	1 ← 1	↑ 1	1 ← 1	↑ 2	2 ← 2	↑ 2
3	M	0 ← 0	↑ 0	1	↑ 2	2 ← 2	↑ 2	2 ← 2	↑ 2
4	K	0	1 ← 1	↑ 1	2	↑ 2	2 ← 2	↑ 3	3 ← 3
5	L	0	1	2 ← 2	↑ 2	2	↑ 3	3 ← 3	↑ 4
6	K	0	1	2	↑ 2	3	↑ 3	4 ← 4	↑ 4

Figure 3.8 The final matrix after filling step of dynamic programming for LCS problem.

The complete part of the matrix filling step is shown in Figure 3.8. After computing a solution to a subproblem, it is stored in a matrix as shown in matrix fill step. This refers memoization technique. Subsequent subproblems check the matrix to avoid redoing work so computational complexity of dynamic programming reduces [42].

TRACEBACK :

Figure 3.9 Traceback step of dynamic programming for LCS problem.

The common elements of two sequences A and B are shown in gray boxes in Figure 3.9. Start is the last position in the matrix and the cost value of last position shows the length of the longest common sequences. In order to find the longest common subsequences, arrows are tracked until arriving the 0th column or row in the matrix. Diagonal arrows show the LCS in the path you track. The possible longest common subsequences are shown in Figure 3.10.

Figure 3.10 All possible longest common subsequences. (a) LCS is LNKL, (b) LCS is LMKL, (c) LCS is LMLK.

3.1.2.1 USAGE OF DP IN STEREO CORRESPONDENCE

A maximum likelihood (ML) stereo algorithm, realized by Ingemar Cox in [21], performs dense stereo correspondence in stereo vision using a maximum likelihood formulation. It presents maximum likelihood cost function which has to be optimized. Cost function models the corresponding points in the left and right images. In real-time stereo images, changes in illumination conditions and differences in camera responses are big problems therefore the maximum likelihood cost function also models these problems. The field of view of each camera, the

probability of each camera imaging a point in the scene and the variance associated with camera sensor noise are represented in the cost function.

Cox [21] investigated and used the sensor fusion methodology of Pattipati [39] in order to show the cost of the matching and occluded features which are related with physical and statistical characteristics of the scene and the cameras. The maximum likelihood cost function which is the combination of matching pixels' costs and occluded pixels' costs must be minimized hence dynamic programming technique is selected because this optimization technique is the best one in order to realize minimization.

First the maximum likelihood cost function derived by Cox [21] will be shown and then dynamic programming solution for minimizing this cost function will be explained.

$$\sum \left\{ \delta_{i1,i2} \ln \left(\frac{P_D^2 \Phi}{(1-P_D) |(2\pi)^{dim} S|^{1/2}} \right) + (1 - \delta_{i1,i2}) [(1/4) (z_{i1} - z_{i2})' S^{-1} (z_{i1} - z_{i2})] \right\} \quad (3.6)$$

Cox's maximum likelihood cost function [21] is shown in Eq (3.6). The first half of the equation represents the occlusion cost and the second half represents the matching cost of the observed pixels. Stereo cameras are denoted by $s=\{1,2\}$. $\delta_{i1,i2}$ is an indicator and equals to zero when the observed pixels are corresponding and equals to one when the observed pixels are not (which means one of them is occluded). P_D represents the probability of each camera imaging a point in the scene and Φ represents the field of view of each camera. The maximum likelihood stereo algorithm of Cox observes the corresponding epipolar scanlines and tries to find whether the observed pixels are corresponding or not by using Eq. (3.6). Therefore, it uses the intensities of pixels which are denoted by z_{i1} and z_{i2} . S represents the covariance matrix associated with the error $(z_{i1} - z_{i2})$ and dim is the dimension of the measurement z_{is} [21].

Dimension (dim) of the searching area is equal to one because only the corresponding epipolar scanlines are searched by ML stereo algorithm. Camera sensor noise affects the result of finding corresponding pixels and means an error in

stereo vision. ML stereo algorithm considers the variance of cameras sensor noise as the inverse of covariance S . Therefore the cost of an occlusion and matching pixels are given below [43] respectively based on Eq. (3.6).

$$\text{Cost of matching (NOC)} = (I_l(x) - I_r(x + d))^2 \sigma^2 / 4$$

$$\text{Cost of occlusion (OC)} = \ln(P_D^2 \Phi) / ((1 - P_D) (2\Pi / \sigma^2)^{1/2})$$

where ;

$I_l(x)$ and $I_r(x + d)$ represent pixel intensities at positions x and $x + d$.

σ^2 represents variance associated with camera sensor noise.

d represents disparity.

P_D representst probability of each camera imaging a point in the scene.

Φ represents field of view of each camera.

The minimization of Eq. (3.6) specifies the decision whether the observed pixels are corresponding pixels or not. Uniqueness and ordering constraints, explained in Chapter 2, are the basic constraints in the ML stereo and the application of them is illustrated in Figure 3.11.

Figure 3.11 A path representing a matching of points in the left and right images [21].

Diagonal path, shown in Figure 3.11, specifies the matching pixels in the left and right epipolar scanlines of stereo images, horizontal path specifies the occluded pixels in the left image and vertical path specifies the occluded pixels in the right image [21]. The dashed path doesn't obey the rules of uniqueness and ordering constraints. Dynamic programming uses the rules of uniqueness and ordering constraints shown in Figure 3.11 and the cost of matching (NOC) and the cost of occlusion (OC) functions for minimizing Eq. (3.6). Now, three steps of dynamic programming in ML stereo algorithm (initialization, matrix fill and traceback) will be analyzed.

INITIALIZATION :

A matrix with N columns and N rows is constructed where N corresponds to the size of the epipolar scanlines in right and left images. The 0th row and column of the matrix can be initially filled with $(i-1)*OC$, where $0 < i \leq N$, shown in Figure 3.12.

		Right Scanline						
		0	1	2	-----			N-1
Left Scanline	0	0	OC	2OC	---	---	---	(N-1) OC
	1	OC						
	2	2OC						
	---	---						
	---	---						
	---	---						
	N-1	(N-1) OC						

Figure 3.12 Initialization step of dynamic programming for ML stereo problem.

MATRIX FILL :

Matrix fill step of dynamic programming in stereo is the most important step. According to the observed pixels in the right and left scanlines, if the value of cost function NOC is smaller than the cost function OC, diagonal path in Figure 3.11 must be tracked and if OC is smaller than NOC, vertical or horizontal path must be tracked so dynamic programming solution models these conditions by creating matching cost function $M_{i,j}$, which is shown below and fills the matrix by obeying the rules of $M_{i,j}$.

$$M_{i,j} = \text{MINIMUM} [M_{i-1,j-1} + \text{NOC (matching, correspondence pixels)}, \\ M_{i,j-1} + \text{OC (occlusion in right image)}, \\ M_{i-1,j} + \text{OC (occlusion in left image)}]$$

Figure 3.13 Matrix fill step of dynamic programming for ML stereo problem.

$$a = M_{1,1} = \min[(M_{0,0}+NOC), (M_{1,0}+OC), (M_{0,1}+OC)]$$

$$b = M_{1,2} = \min[(M_{0,1}+NOC), (M_{0,2}+OC), (a+OC)]$$

$$c = M_{2,1} = \min[(M_{1,0}+NOC), (M_{2,0}+OC), (a+OC)]$$

While calculating the value $M_{i,j}$ for position (i,j) , if the value of $(M_{i-1, j-1}+NOC)$ is minimum then diagonal arrow must be set for position (i,j) which means that i^{th} pixel in the right image and j^{th} pixel in the left image are matched. If the value of $(M_{i-1, j}+OC)$ is minimum then vertical arrow must be set for position (i,j) which means that i^{th} pixel in the left image is occluded. If the value of $(M_{i, j-1}+OC)$ is minimum then horizontal arrow must be set for position (i,j) which means that j^{th} pixel in the right image is occluded.

TRACEBACK :

After filling the matrix, matching and occluded pixels can be determined by starting the last position $(N-1, N-1)$ and tracking the dimensions of arrows. In the path you track, diagonal arrows specify the matching pixels, and others specify the occluded pixels. Tracking is finished when you arrive the 0^{th} column or row.

3.2 IMAGE SEGMENTATION ALGORITHM

3.2.1 MEAN SHIFT SEGMENTATION

Mean-shift (MS) is an iterative nonparametric clustering approach that widely used in pattern recognition tasks. [44] uses MS technique for analyzing the feature space of images and realizes two low-level vision tasks, discontinuity preserving filtering and image segmentation. Image segmentation part of [44] will be described and explained in this section.

Feature space can be regarded as the empirical probability density function (p.d.f) of the represented parameter. That's why kernel density estimation which is the way of p.d.f estimation, is used for analyzing feature space.

Given n data points x_i , $i = 1, \dots, n$ in the d dimensional space R^d , the kernel density estimator with kernel $K(x)$ and bandwidth (smoothing parameter) h , is given in Eq. (3.7).

$$f(x) = \frac{1}{n h^d} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (3.7)$$

Radially symmetric kernels which satisfy the condition in Eq. (3.8), are used in [44].

$$K(x) = c_{k,d} k(|x|^2) \quad (3.8)$$

$c_{k,d}$ is the normalization constant and $k(x)$ is the profile of the kernel. Eq. (3.7) can be rewritten as Eq. (3.9).

$$f(x) = \frac{c_{k,d}}{n h^d} \sum_{i=1}^n k\left(\left|\frac{x - x_i}{h}\right|^2\right) \quad (3.9)$$

Gradient of density estimator is equal to zero in the modes of density and this is the first step for analyzing feature space. MS procedure locates these zeros without estimating the density which is shown in [44] in detail. MS vector points towards the direction of the maximum increase in the density and stops when the value of y which is shown in Eq. (3.10), converges. Convergence means that preceding and current value of y are equal.

$$y_{j+1} = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{y_j - x_i}{h}\right\|^2\right)} \quad j=1,2,\dots \quad (3.10)$$

where $g(x) = -k'(x)$

MS vector and its convergence are shown in Figure 3.14. The window of the kernel is centered at pixel x in the image and is shifted by MS vector. Arrows represent the MS vectors in Figure 3.14. For each shifted window, density estimator is calculated to determine the MS vector and convergence value y is computed and checked.

Figure 3.14 MS vector and its convergence.

Before constructing image segmentation, color space is initially mapped to another color space like L^*u^*v or L^*a^*b because color differences should

correspond to Euclidean distances in the color space. Euclidean metric is not guaranteed for a color space. L means the lightness coordinate and others mean chromaticity coordinates [44].

An image which is defined as a two dimensional lattice of p dimensional vectors (pixels) where p=1 for gray-level case and p=3 for color images, has two fundamental informations in spatial domain and range domain. The spatial domain presents the space of the lattice and the range domain presents gray level, color or spectral information of the image. [44] uses the multivariate kernel, shown in Eq. (3.11), which is the product of two radially symmetric kernels for modeling the spatial and range domains [44].

$$K_{hs,hr}(x) = \frac{C}{h_s^2 h_r^p} k\left(\left\|\frac{x^s}{h_s}\right\|^2\right) k\left(\left\|\frac{x^r}{h_r}\right\|^2\right) \quad (3.11)$$

Where x^s is the spatial part of a feature vector
 x^r is the range part of a feature vector
 $k(x)$ is the profile of the kernel
 h_s and h_r are the kernel bandwidths
 C is the normalization constant

Epanechnikov kernel profile has always a good performance and is generally used in practice.

Image segmentation which uses mean shift procedure [44], has four steps which are listed below:

1. For each pixel x_i , $i = 1, 2, \dots, n$, in the image, initialize $j=1$ and $y_{i,1} = x_i$. Then compute $y_{i,j+1}$ according to Eq. (3.10) until convergence, $y = y_{i,c}$. Finally assign the value of convergence $y_{i,c}$ to the pixel x_i . The new value of pixel (z_i) is equal to $(x_i^s, y_{i,c}^r)$. This operation is also known as mean shift filtering.

2. The clusters $\{C_p\}_{p=1,\dots,v}$ are formed by grouping the new values of pixels (z_i) which are closer than h_s in the spatial domain and h_r in the range domain.
3. Labeling the pixels according to the clusters.
4. Eliminate spatial regions containing less than M pixels.

Here, the resolution of the analysis can be set by changing h_s and h_r values and also the minimum region size can be set by M .

CHAPTER 4

FUSION OF STEREO ALGORITHMS FOR ACCURATE DISPARITY ESTIMATION AND DISPARITY SEGMENTATION

4.1 INTRODUCTION

The block diagram of the method used in this thesis is given in Figure 4.1. The three stereo algorithms (SAD correlation-based stereo, dynamic programming and region-based stereo) are implemented and fused in order to obtain high accurate disparity map and then disparity map is segmented for the purpose of object detection by using mean shift segmentation. Various improvement methods are applied at various steps of our method.

Figure 4.1 The proposed methodology.

In the rest of this chapter, three stereo algorithms, disparity segmentation and various improvement methods are explained.

4.2 STEREO ANALYSIS BLOCK

4.2.1 SAD CORRELATION-BASED STEREO

One of the implemented stereo algorithms is SAD correlation-based stereo which has been explained in section 3.1.1. Four individual algorithms are implemented in this section. The first one named as *SAD* finds the disparity map of the stereo images by using only a fixed sized window and SAD correlation cost function. The second one named as *SAD_CON* adds the left-right consistency check technique to *SAD* algorithm so it can label the occluded pixels. The third one named as *SAD_MW* uses the *SAD* algorithm with multiple windows. The last one named as *SAD_MW_CON* uses *SAD_MW* algorithm with left-right consistency check technique. In other words, it combines two techniques (left-right consistency check and multiple window) with *SAD* algorithm. By using these four algorithms, effects of the left-right consistency check and the multiple window techniques can be observed easily. Comparison of these four algorithms and effects of some parameters such as window size are tested in Chapter 5.

4.2.2 DYNAMIC PROGRAMMING

Dynamic programming (optimization technique) is used in stereo vision for obtaining disparity map of stereo images which has been explained in section 3.1.2.1 in detail. Comparison of this algorithm with others and effects of some parameters such as the variance of camera sensor noise are tested in Chapter 5.

4.2.3 REGION-BASED STEREO

Region-based stereo aims to segment the reference image into small regions by using mean shift segmentation algorithm which is explained in section 3.2.1 and then matches these regions with the regions on the other image. Matching process is realized by using sum of absolute difference correlation similarity of pixels in the regions. The disparity values of the pixels in the same region are supposed to be same. For finding the occluded pixels, the intersection of matching regions on the other image are observed and one of the candidate pixels in the reference image is specified as an occluded by using minimum correlation similarity. For example, there are two regions R_1 (green) and R_2 (gray) in the left image (reference image) as shown in Figure 4.2. The matching regions are R_1' and R_2' in the right image. Here, the problem is whether the pixels of intersection region U (yellow) match with the pixels in A or the pixels in B. This problem can be solved with SAD correlation similarity for each pixel in U . If SAD of a pixel is minimum with a pixel in A then the candidate pixel in B is specified as occluded or vice versa.

Figure 4.2 Region-based stereo.

There are two types of region-based algorithm which are purely region-based (*Region_based*) and region-based with finding occluded pixels (*Region_based_with_occlusion*). Comparisons of these algorithms are given in Chapter 5.

4.2.4 FUSION OF STEREO ALGORITHMS

Fusion of stereo algorithms is an alternative technique in stereo vision. The disparity maps obtained by three stereo algorithms (SAD correlation-based stereo, dynamic programming and region-based stereo) are analyzed and if the disparity values of a pixel are same in the disparity maps of fusion algorithms then that disparity value is considered as true. If the disparity values of a pixel are different among the disparity maps of fusion algorithms then that disparity is supposed to be unknown and the disparity value of that pixel is equalized to zero for demonstration purpose. For example, let us define two stereo algorithms (algorithm 1 and algorithm 2) and their disparity maps as shown in Figure 4.3 (a) and (b). The fusion of algorithm 1 and algorithm 2 are also demonstrated in Figure 4.3 (c). The same disparity values of a pixel are conserved and the unknown disparity values are shown as U in the disparity map of fusion.

Figure 4.3 An example of fusion. (a) Disparity map of algorithm 1, (b) Disparity map of algorithm 2, (c) Disparity map of fusion.

The unknown pixels can be assigned some values by using improvement algorithm. Improvement for assigning disparity values to unknown disparities is explained in the next subsection and the comparisons of the fusion algorithms are given in Chapter 5.

4.2.5 IMPROVEMENT

This is a disparity refinement approach. The disparity values of unknown pixels which are labelled as a result of the fusion algorithms can be found using continuity constraint which is explained in section 2.1.1. Continuity constraint states that the disparity of the corresponding pixels varies smoothly almost everywhere over the image. Pixels on object boundaries are the exceptions where the continuity constraint fails because disparity changes abruptly at boundaries. Therefore, the disparity of a matched pair and that of its nearest neighbor pair should be alike if they are not at the boundaries. So, the unknown pixels existing between the disparity values whose differences are not bigger than 1, can be assigned a value which is the average of its neighbors. This is like a filtering by using an average filter. The effects of improvement on the results of the fusion algorithms are given in Chapter 5.

4.3 DISPARITY SEGMENTATION BLOCK

The block diagram of disparity segmentation is given in Figure 4.4. First disparity map obtained from the stereo images is processed for determining separated depth region if there exists any. The regions which are located in different disparity levels and far away from each other are called as separated depth regions. Because the values of pixels specify the depth value, separated depth regions can be detected by analyzing the pixels of disparity map. Finding the separated depth regions facilitates the segmentation process since then segmentation can be applied only to a smaller region. If the objects are closer to each other in the stereo images, strictly

separated depth regions can not be found that's why the whole disparity image is investigated for segmentation.

Figure 4.4 Disparity segmentation method .

Before applying the actual disparity segmentation, the disparity values of pixels are scaled by a scale factor which is determined as follows:

$$Scale\ factor = 255 / (Max_Disparity) \quad (4.1)$$

Max_Disparity means the maximum possible disparity value for the pixels in the image. In other words, it defines the search area for finding the matching points in the stereo images. *Scale factor* is used in depth segmentation block to determine depth regions. The pixels of disparity map are firstly sorted and assigned to regions according to a *region_threshold* value related to the *scale factor*. In the algorithms used in this thesis, *region_threshold* is generally set to half of the *scale factor*. But in some images having more than one object which are located closer to each other, *region_threshold* may be set to smaller values as shown in Chapter 5.

In the mean shift segmentation, the minimum segmentation area (*Min_Seg_Area*) should be set. It defines the minimum number of pixels which should exist in each segmented region. In this thesis, the default value of minimum segmentation area is set to 5000 for 250x320 sized images.

Results of the disparity segmentation block for different images are given in Chapter 5.

At the beginning of the disparity segmentation block, mean shift segmentation is not applied to the whole disparity map because mean shift segmentation has not good results in the depth regions which are back to back and closer to each other. That's why the method of investigating the seperated depth regions which is shown in Figure 4.4 is used. The results of applying mean shift segmentation to the whole disparity map are also shown in Chapter 5.

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1 STEREO VISION TEST PROCEDURE

There are three basic stereo algorithms named as SAD correlation-based stereo, dynamic programming and region-based stereo which are explained in Chapter 3 and Chapter 4. In order to evaluate the performance of these stereo algorithms or the effects of varying some of their parameters, methods defined in the taxonomy paper [16] is used. [16] has two quality measures which are RMS (root-mean-squared) error and percentage of bad matching pixels. Some stereo image data sets (Tsukuba, Venus and Sawtooth) [16], [48] shown in Appendix which have ground truth disparity maps are tested using RMS and bad matching pixel error measurements.

RMS ERROR :

RMS error is defined in Eq. (5.1).

$$R = \left(\frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_T(x,y)|^2 \right)^{\frac{1}{2}} \quad (5.1)$$

where $d_C(x,y)$ is the estimated disparity value for pixel (x,y) and $d_T(x,y)$ is the ground truth value and N is the total number of pixels.

PERCENTAGE OF BAD MATCHING PIXELS :

Percentage of bad matching pixels is defined in Eq. (5.2).

$$B = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d) \quad (5.2)$$

δ_d is a disparity error tolerance and equals to 1 in evaluating bad matching pixels of stereo algorithms.

There are three important regions in stereo images. These are occluded, textureless and depth discontinuity regions. Determining RMS error and percentage of bad matching pixels are important in these regions and in [16], [48], these regions are investigated separately. For example occluded/non-occluded, textureless/textured and depth discontinuity regions of Sawtooth are shown in Figure 5.1.

Figure 5.1 Different regions of Sawtooth data set. (a) Original image, (b) Ground truth disparity map, (c) Occluded regions (black) and non-occluded regions (white), (d) Textureless regions (black) and textured regions (white), (e) Depth discontinuity regions (black) [48].

In the following experiments, only the percentage of bad matching pixels is used for comparing stereo algorithms. The quality measures which are mostly used in the following performance comparisons are denoted in Table 5.1.

Table 5.1 Quality measures of stereo images.

Quality Measures	Description
Bo	Percentage of bad pixels in the occluded regions
Bo*	Percentage of bad pixels in the non-occluded regions
Bt	Percentage of bad pixels in the textured regions
Bt*	Percentage of bad pixels in the textureless regions
Bd	Percentage of bad pixels in the depth discontinuity regions

The performances of SAD Correlation-based, dynamic programming and region-based stereo algorithms are determined respectively in the following subsections according to the quality measures.

5.2 RESULTS OF STEREO ALGORITHMS APPLIED TO STEREO DATA SETS

5.2.1 SAD CORRELATION-BASED STEREO

SAD correlation-based stereo is explained in Chapter 3 and Chapter 4. Four versions of correlation-based algorithms (where left-right consistency check and/or multiple window supporting techniques are used) are tested. These are *SAD*, *SAD_CON*, *SAD_MW* and *SAD_MW_CON* which are explained in Chapter 4.

In the tests, 9x9 correlation window is used. The percentage of bad matching pixels (results of correlation-based algorithms) are given in Table 5.2 for five quality measures mentioned in Table 5.1 and the disparity maps are given in Figure 5.2 for Tsukuba, Venus and Sawtooth data sets. The performance ranking of the algorithms is also written as superscript for each quality measure.

Table 5.2 The percentage of bad matching pixels of four versions of SAD correlation-based algorithm for five quality measures for three stereo data sets.

TSUKUBA IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
SAD	100 ⁽³⁾	23.8 ⁽³⁾	50.2 ⁽³⁾	20.2 ⁽¹⁾	26.4 ⁽³⁾
SAD_CON	55.8 ⁽²⁾	16.5 ⁽¹⁾	34.0 ⁽²⁾	23.7 ⁽²⁾	11.3 ⁽¹⁾
SAD_MW	100 ⁽³⁾	26.7 ⁽⁴⁾	51.4 ⁽⁴⁾	24.0 ⁽³⁾	28.7 ⁽⁴⁾
SAD_MW_CON	35.0 ⁽¹⁾	20.2 ⁽²⁾	25.2 ⁽¹⁾	31.0 ⁽⁴⁾	12.7 ⁽²⁾
VENUS IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
SAD	100 ⁽³⁾	15.1 ⁽²⁾	45.0 ⁽⁴⁾	20.1 ⁽¹⁾	12.6 ⁽³⁾
SAD_CON	76.5 ⁽²⁾	12.8 ⁽¹⁾	40.0 ⁽²⁾	21.6 ⁽²⁾	8.4 ⁽¹⁾
SAD_MW	100 ⁽³⁾	18.8 ⁽⁴⁾	44.7 ⁽³⁾	26.1 ⁽³⁾	15.1 ⁽⁴⁾
SAD_MW_CON	54.9 ⁽¹⁾	18.1 ⁽³⁾	28.8 ⁽¹⁾	31.7 ⁽⁴⁾	11.2 ⁽²⁾
SAWTOOTH IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
SAD	100 ⁽³⁾	14.5 ⁽³⁾	44.7 ⁽³⁾	11.9 ⁽³⁾	14.7 ⁽³⁾
SAD_CON	46.9 ⁽²⁾	7.4 ⁽²⁾	33.4 ⁽²⁾	5.8 ⁽¹⁾	7.5 ⁽²⁾
SAD_MW	100 ⁽³⁾	15.1 ⁽⁴⁾	45.5 ⁽⁴⁾	12.9 ⁽⁴⁾	15.4 ⁽⁴⁾
SAD_MW_CON	29.0 ⁽¹⁾	7.0 ⁽¹⁾	23.1 ⁽¹⁾	6.3 ⁽²⁾	7.0 ⁽¹⁾

In Table 5.2, the percentage of bad matching pixels is observed in five different regions. The superscript numbers indicate the rank of each algorithm in each column. Although the algorithms perform differently for the three different data sets, we can comment about their general behavior. Venus image pair includes slanted surfaces, Sawtooth image pair includes textureless regions and Tsukuba

image pair is the most complicated one and includes many objects at different disparity layers.

SAD and *SAD_MW* algorithms are not considered at occluded regions because they can not find occluded pixels.

According to the results, *SAD_MW_CON* algorithm is the best one at occluded and depth discontinuity regions for each data set because left-right consistency check and multiple supporting windows methods improve the performance of correlation-based algorithms at the boundaries of objects where depth discontinuity and occluded regions exist [37].

At non-occluded and textured regions, *SAD_CON* have better performance for Venus and the performances of *SAD_MW_CON* and *SAD_CON* algorithms are close to each other and better than others for Tsukuba and Sawtooth.

At textureless region, *SAD_CON* and *SAD_MW_CON* algorithms have better performance for Sawtooth. *SAD* algorithm have better performance for Tsukuba and Venus at textureless region but the values of bad matching pixels of all algorithms are close to each other at textureless region for Tsukuba and Venus.

Consequently, the improvement techniques of correlation (left-right consistency check and multiple supporting windows) have the best performance at occluded and depth discontinuity regions. They also do not decrease the performance of correlation algorithms significantly at other regions like textureless, textured and non-occluded.

In order to determine the performance of correlation-based algorithms at all regions for each of the data set, the average rank value of all regions can be computed for the specified algorithm. Average rank value can be calculated by using Eq. (5.3).

$$\text{Average rank value} = \left(\sum_{i=1}^N R(i) \right) / N \quad (5.3)$$

where $R(x)$ = Rank value of x region
 N = Total no. of regions

The superscript numbers in Table 5.2 indicate the rank value (R) of each algorithm in each column and the total number of regions (N) is five (occluded, non-occluded, depth discontinuity, textureless, textured). Algorithm which has minimum average rank value is the best one for specified data set. Table 5.3 presents the average rank value of correlation-based algorithms for each data set.

Table 5.3 The average rank value of correlation-based algorithms for each data set.

AVERAGE RANK VALUES			
ALGORITHM	TSUKUBA	VENUS	SAWTOOTH
SAD	2.6	2.6	3
SAD_CON	1.6	1.6	1.8
SAD_MW	3.6	3.4	3.8
SAD_MW_CON	2	2.2	1.2

According to Table 5.3, *SAD_MW_CON* has the best performance for Sawtooth and *SAD_CON* has the best performance for Tsukuba and Venus.

In Figure 5.2, left image of the stereo pair is selected as the reference image so disparity maps are constructed according to the left image. In the disparity maps of *SAD_CON* and *SAD_MW_CON* algorithms, occluded pixels are shown as black. Generally the left borders of objects in the images are black because these regions are generally occluded in the right image.

In the rest of the chapter, left image will always be used as the reference image and disparity maps are constructed according to the left images.

Figure 5.2 Outputs of SAD correlation-based algorithms for Tsukuba, Venus and Sawtooth. (a) Ground truths, (b) SAD, (c) SAD_CON, (d) SAD_MW, (e) SAD_MW_CON.

In *SAD* correlation algorithms, window size is an important factor for performance. The effect of window size can also be evaluated based on the quality measures. *SAD_MW_CON* algorithm is used for testing the effect of window size and the results are shown in Table 5.4.

Table 5.4 The effect of window size on correlation-based stereo.

TSUKUBA IMAGE PAIR					
Window Size	Bo	Bo*	Bd	Bt*	Bt
3x3	30.2	46.8	42.6	63.1	35.3
9x9	35.0	20.2	25.2	31.0	12.7
15x15	56.5	13.8	27.7	19.4	9.8
21x21	63.4	11.0	30.5	13.5	9.3
27x27	73.5	10.5	33.5	10.9	10.2
51x51	86.9	12.9	41.2	12.5	13.1
71x71	87.1	13.8	44.2	12.6	14.6
101x101	83.6	15.9	48.1	13.7	17.4
VENUS IMAGE PAIR					
Window Size	Bo	Bo*	Bd	Bt*	Bt
3x3	47.3	51.9	50.2	75.9	39.8
9x9	54.9	18.1	28.8	31.7	11.2
15x15	72.0	9.9	33.0	15.6	7.0
21x21	75.4	6.9	35.6	9.7	5.6
27x27	81.7	6.2	37.6	7.9	5.3
51x51	86.6	7.1	39.5	8.9	6.2
71x71	89.8	9.5	39.2	12.2	8.2

Table 5.4 Continued.

101x101	87.9	13.6	42.7	16.6	12.1
SAWTOOTH IMAGE PAIR					
Window Size	Bo	Bo*	Bd	Bt*	Bt
3x3	30.1	35.5	38.3	61.4	32.8
9x9	29.0	7.0	23.1	6.3	7.0
15x15	50.2	5.5	25.3	3.5	5.7
21x21	50.2	5.3	29.3	4.3	5.4
27x27	66.5	5.6	29.9	4.6	5.7
51x51	82.3	7.4	32.6	7.8	7.3
71x71	88.9	9.7	35.8	11.8	9.4
101x101	94.2	12.8	42.0	15.2	12.6

To better understand the effect of window size, the graph of the bad pixels versus window size is presented in Figure 5.3 for various regions. According to Figure 5.3, percentage of bad pixels at occluded regions increases when the size of window increases. At other regions, smaller or bigger window size are not suitable for calculating the true disparities of the pixels. Moreover, big window size increases the computational complexity of the algorithms.

Generally Bo*, Bd and Bt* quality measures are considered for comparing the algorithms in [16] therefore window size between 9x9 and 15x15 may be useful in order to get good results at all regions.

Figure 5.3 The effect of window size on correlation-based stereo (a) B_o versus window size, (b) B_o^* versus window size, (c) B_t^* versus window size, (d) B_t versus window size, (e) B_d versus window size.

5.2.2 DYNAMIC PROGRAMMING

Dynamic programming in stereo is explained in Chapter 3 and Chapter 4. There are three variables in dynamic programming which are the variance of camera sensor noise (σ^2), the probability of each camera imaging a point in the scene (P_D) and the field of view of each camera (Φ). ($1-P_D$) may be viewed as the probability of occlusion [21] so P_D can be selected as 0.99 and also Φ can be selected as 180° (the maximum values for P_D and Φ) in the test shown below. Camera sensor noise is accepted as white noise. The effect of variance of camera sensor noise in the data sets and its optimal value are presented in Table 5.5.

Table 5.5 The effect of the variance of camera sensor noise on dynamic programming in stereo.

TSUKUBA IMAGE PAIR					
Variance	Bo	Bo*	Bd	Bt*	Bt
36	8.8	76.0	83.2	61.8	86.0
25	14.2	62.9	72.2	45.3	75.3
16	20.1	53.7	62.1	36.1	66.1
9	28.1	39.0	46.5	24.0	49.6
4	34.3	26.1	35.5	12.4	35.7
1	40.1	14.8	26.1	6.3	20.7
0.5	43.4	12.0	23.7	6.1	16.1
0.1	51.1	8.1	20.1	7.0	8.9
0.05	55.0	8.6	20.4	9.7	7.9
0.01	67.1	12.8	27.8	17.4	9.6
0.001	85.7	19.3	38.5	25.6	14.9

Table 5.5 Continued.

VENUS IMAGE PAIR					
Variance	Bo	Bo*	Bd	Bt*	Bt
36	12.9	74.6	81.4	67.7	78.1
25	18.6	63.6	74.2	54.0	68.4
16	22.0	57.0	69.9	45.9	62.7
9	28.1	43.4	58.3	33.0	48.6
4	31.8	29.0	48.5	14.6	36.3
1	42.9	16.3	37.1	6.1	21.5
0.5	46.3	13.8	33.8	7.6	16.9
0.1	57.0	10.6	30.8	10.6	10.6
0.05	61.4	11.2	31.3	13.0	10.3
0.01	70.7	15.7	33.5	21.8	12.6
0.001	78.9	30.6	43.7	42.5	24.6
SAWTOOTH IMAGE PAIR					
Variance	Bo	Bo*	Bd	Bt*	Bt
36	3.0	87.4	96.3	66.8	89.5
25	7.1	80.7	92.5	51.0	83.8
16	10.5	74.2	88.7	42.1	77.5
9	15.2	60.3	80.4	26.4	63.8
4	25.6	44.5	66.2	11.5	47.8
1	41.3	21.8	42.1	3.1	23.7
0.5	45.0	16.2	36.5	3.3	17.5
0.1	57.7	9.1	29.2	5.7	9.4
0.05	64.2	8.2	27.6	7.4	8.3
0.01	78.1	7.1	26.7	9.1	6.9
0.001	93.1	9.1	31.5	10.3	8.9

According to Table 5.5, the percentage of bad pixels at occluded regions decreases when the variance of camera sensor noise increases because the cost of matching (NOC) increases much more than the cost of occlusion (OC). So, there are a lot of pixels which are labelled as occluded and that is why the probability of bad matching pixel selection in occluded region decreases.

At other regions, percentage of bad matching pixels decreases until a specified variance value. This value exists between 1 and 0.01 for all regions for each data set. So, the value of variance can be chosen as 0.1 in the following subsections. Some of the disparity maps for different variance values are shown in Figure 5.4.

Dynamic programming has horizontal streaks in the disparity map, especially in regions of low texture. The horizontal streaks can be easily seen in Figure 5.4 (e) and (f). These streaks occur because all horizontal scanlines are matched independent from each other in dynamic programming. For solving streak problem, dynamic programming can be applied twice (the first time, the horizontal lines are matched from left to right, and afterwards right to left) [49]. Some vertical optimization techniques can be used with dynamic programming and subpixel accuracy and some noise reduction techniques which use filtering can be used for handling streak problem [49]. These solutions are explained in [49] and they have satisfactory results but they generally have high computational complexity. Therefore they are not used in this thesis.

Figure 5.4 Effect of the camera sensor noise variance on DP for Tsukuba, Venus and Sawtooth. (a) Ground truths, (b) Variance=25, (c) Variance=9, (d) Variance=1, (e) Variance=0.1, (f) Variance=0.01.

5.2.3 REGION-BASED STEREO

There are two types of region-based algorithm, *Region_based* and *Region_based_with_occlusion* which are explained in Chapter 4. The percentage of bad matching pixels for these algorithms is listed in Table 5.6 and Figure 5.5 for each stereo pair.

Table 5.6 The percentage of bad matching pixels of region-based algorithms for five quality measures for each stereo data set.

TSUKUBA IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
Region_based	100	4.5	13.2	2.31	6.0
Region_based _with_occlusion	54.1	6.2	15.4	3.3	8.2
VENUS IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
Region_based	100	17.0	29.4	20.1	15.4
Region_based _with_occlusion	66.7	17.9	30.1	20.6	16.5
SAWTOOTH IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
Region_based	100	5.3	12.1	10.1	4.8
Region_based _with_occlusion	53.3	6.4	14.6	10.5	6.0

Figure 5.5 Outputs of region-based algorithms for Tsukuba, Venus and Sawtooth. (a) Ground truths, (b) *Region_based*, (c) *Region_based_with_occlusion*.

In *Region_based_with_occlusion* algorithm, *Region_based* algorithm is run first and then the occluded pixels are detected. In other words these algorithms are nearly the same. However, according to Table 5.6, it can be seen that *Region_based_with_occlusion* algorithm is worse than *Region_based* algorithm in regions other than occluded region because while finding occluded pixels, some false selections occur in those regions. *Region_based* algorithm is not used in the following comparisons and processes. It is mentioned here only for comparison with *Region_based_with_occlusion* algorithm.

Most of the stereo matching algorithms (especially pixel-based) fail in sufficiently large regions that are weakly textured. On the other hand, region-based algorithms can handle this problem and have high performance in large homogeneous regions. For example, the lamp of Tsukuba image has large textureless region and the performance of region-based algorithm is satisfactory in this region as seen in Figure 5.5. Furthermore, region-based algorithms generally allow the accurate localization of object boundaries which have large textureless region.

5.2.4 FUSION OF STEREO ALGORITHMS AND IMPROVEMENT

Fusion of stereo algorithms is explained in Chapter 4. Only the stereo algorithms which can also find occluded pixels are considered for fusion. These are *SAD_MW_CON*, *SAD_CON*, dynamic programming and *Region_based_with occlusion*. For SAD correlation-based algorithms, 9x9 windows are used and variance of camera sensor noise is selected as 0.1 for dynamic programming. The fused algorithms are named and listed in Table 5.7.

As a result of fusion process, there can be some pixels which are not assigned disparity values in the disparity map. These unknown disparities can be determined by the improvement step. Improvement step is explained in Chapter 4.

The percentage of bad matching pixels of fusion and improved fusion algorithms are presented together in Table 5.8. Disparity maps of fusion and improved fusion algorithms are presented together in Figure 5.6.

Table 5.7 The fused algorithms and their labels.

FUSION LABEL	FUSED ALGORITHMS		
SAD_MW_CON & DP	<i>SAD_MW_CON</i>	Dynamic programming	-
Region & SAD_MW_CON	<i>Region_based_with occlusion</i>	SAD_MW_CON	-
Region & DP	<i>Region_based_with occlusion</i>	Dynamic programming	-
Region & SAD_MW_CON & DP	<i>Region_based_with occlusion</i>	SAD_MW_CON	Dynamic programming
SAD_CON & DP	<i>SAD_CON</i>	Dynamic programming	-
Region & SAD_CON	<i>Region_based_with occlusion</i>	<i>SAD_CON</i>	-
Region & SAD_CON & DP	<i>Region_based_with occlusion</i>	<i>SAD_CON</i>	Dynamic programming

Table 5.8 The percentage of bad matching pixels of fusion algorithms and improved fusion algorithms for five quality measures for each data set.

TSUKUBA IMAGE PAIR										
ALGORITHM	Bo		Bo*		Bd		Bt*		Bt	
	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.
SAD_MW_CON & DP	64.1	41.557	37.8	5.509	40.5	22.894	56.5	5.330	24.6	5.635
Region & SAD_MW_CON	68.5	38.570	38.7	5.636	39.5	23.313	55.4	4.090	26.9	6.724
Region & DP	72.9	47.397	27.9	5.104	39.5	22.922	31.5	3.182	25.4	6.457
Region & SAD_MW_CON & DP	80.1	41.240	45.5	6.710	48.5	28.777	62.2	5.268	33.7	7.724
SAD_CON & DP	76.2	52.965	34.1	7.044	47.6	30.700	48.4	6.114	24.1	7.698
Region & SAD_CON	77.6	50.068	34.2	6.976	46.4	30.708	46.3	4.690	25.7	8.584
Region & SAD_CON & DP	85.8	47.533	41.9	8.147	54.2	36.307	54.7	6.127	32.9	9.569

Table 5.8 Continued.

VENUS IMAGE PAIR										
ALGORITHM	Bo		Bo*		Bd		Bt*		Bt	
	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.
SAD_MW_CON & DP	72.8	81.920	34.0	3.423	46.1	29.364	47.2	3.655	27.4	3.305
Region & SAD_MW_CON	78.5	76.904	44.7	7.414	49.4	31.524	60.1	10.570	37.0	5.820
Region & DP	79.0	88.916	41.3	8.395	55.7	31.761	46.3	11.383	38.8	6.884
Region & SAD_MW_CON & DP	83.3	74.861	51.9	6.619	59.3	36.886	65.1	9.650	45.3	5.088
SAD_CON & DP	84.6	76.842	30.9	3.768	52.5	36.082	39.2	4.012	26.7	3.644
Region & SAD_CON	84.9	74.985	42.0	6.538	55.5	39.560	54.1	9.049	35.9	5.268
Region & SAD_CON & DP	88.2	69.721	50.3	6.544	63.3	41.352	60.6	9.103	45.0	5.251
SAWTOOTH IMAGE PAIR										
ALGORITHM	Bo		Bo*		Bd		Bt*		Bt	
	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.	Fus.	Imp.
SAD_MW_CON & DP	65.5	58.927	20.0	4.398	46.2	23.769	20.2	3.200	20.0	4.520
Region & SAD_MW_CON	62.8	37.495	23.2	3.653	38.8	22.175	29.1	2.524	22.7	3.768
Region & DP	71.3	55.274	25.4	4.194	47.1	21.428	24.1	4.649	25.5	4.148
Region & SAD_MW_CON & DP	75.7	46.180	30.7	4.689	53.8	26.974	33.8	4.042	30.4	4.755
SAD_CON & DP	74.5	58.107	20.6	5.214	52.8	30.707	18.6	4.311	20.8	5.306
Region & SAD_CON	70.2	45.509	22.8	4.702	46.1	31.169	27.1	3.686	22.4	4.806
Region & SAD_CON & DP	81.2	47.521	30.8	5.469	59.3	34.256	31.8	5.091	30.7	5.508

(d1)

(d2)

(e1)

(e2)

(f1)

(f2)

Figure 5.6 Outputs of fusion and improved fusion algorithms for Tsukuba, Venus and Sawtooth. (a) Ground truths, (b1) SAD_MW_CON & DP, (b2) Improved SAD_MW_CON & DP, (c1) Region & SAD_MW_CON, (c2) Improved Region & SAD_MW_CON, (d1) Region & DP, (d2) Improved Region & DP, (e1) Region & SAD_MW_CON & DP, (e2) Improved Region & SAD_MW_CON & DP, (f1) SAD_CON & DP, (f2) Improved SAD_CON & DP, (g1) Region & SAD_CON, (g2) Improved Region & SAD_CON, (h1) Region & SAD_CON & DP, (h2) Improved Region & SAD_CON & DP.

In fusion process, the unknown disparities are equalized to zero for demonstration purpose which can be seen in Figure 5.6. According to Table 5.8, we can say that the percentage of bad matching pixels of fusion algorithms is large in all regions because there are many pixels which are not assigned values (unknown disparities) as a result of fusion algorithms as seen in Figure 5.6. Unknown disparities are effectively estimated by applying improvement step which can be also seen in Table 5.8 and Figure 5.6.

According to Figure 5.6, we can say that the streak problem exists in the fusion of DP algorithm and this problem is mostly decreased by applying improvement step. Also the boundaries of objects are protected during the estimation of unknown disparities by the improvement step. The unnecessary smoothing effect of improvement step may exist in some places in the disparity map so the exact shape of the objects can not be seen. This is the disadvantage of the improvement step.

The percentage of benefit of improvement step is presented in Table 5.9 for five regions (occluded, nonoccluded, depth discontinuity, textureless, textured) for each data set.

Table 5.9 The percentage of benefit of improvement step for five regions for each data set.

TSUKUBA IMAGE PAIR					
ALGORITHM	OCCLUDED REGION	NON- OCCLUDED REGION	DEPTH DISCONTINUITY REGION	TEXTURELESS REGION	TEXTURED REGION
SAD_MW_CON & DP	35.16849	85.42593	43.4716	90.56637	77.0935
Region & SAD_MW_CON	43.69343	85.43669	40.97975	92.61733	75.00372
Region & DP	34.98354	81.70609	41.96962	89.89841	74.57874
Region & SAD_MW_CON & DP	48.51436	85.25275	40.66598	91.53055	77.08012
SAD_CON & DP	30.49213	79.34311	35.5042	87.36777	68.05809
Region & SAD_CON	35.47938	79.60234	33.81897	89.87041	66.59922
Region & SAD_CON & DP	44.60023	80.55609	33.01292	88.7989	70.91489

Table 5.9 Continued.

VENUS IMAGE PAIR					
ALGORITHM	OCCLUDED REGION	NON-OCCLUDED REGION	DEPTH DISCONTINUITY REGION	TEXTURELESS REGION	TEXTURED REGION
SAD_MW_CON & DP	0	89.93235	36.30369	92.25636	87.93796
Region & SAD_MW_CON	2.033121	83.41387	36.18623	82.41265	84.27027
Region & DP	0	79.67312	42.97846	75.41469	82.25773
Region & SAD_MW_CON & DP	10.13085	87.24663	37.79764	85.17665	88.76821
SAD_CON & DP	9.170213	87.80583	31.27238	89.76531	86.35206
Region & SAD_CON	11.67845	84.43333	28.72072	83.27357	85.32591
Region & SAD_CON & DP	20.95125	86.99006	34.67299	84.97855	88.33111
SAWTOOTH IMAGE PAIR					
ALGORITHM	OCCLUDED REGION	NON-OCCLUDED REGION	DEPTH DISCONTINUITY REGION	TEXTURELESS REGION	TEXTURED REGION
SAD_MW_CON & DP	10.03511	78.01	48.55195	84.15842	77.4
Region & SAD_MW_CON	40.29459	84.25431	42.84794	91.32646	83.40088
Region & DP	22.47686	83.48819	54.50531	80.70954	83.73333
Region & SAD_MW_CON & DP	38.99604	84.72638	49.86245	88.04142	84.35855
SAD_CON & DP	22.00403	74.68932	41.8428	76.82258	74.49038
Region & SAD_CON	35.17236	79.37719	32.38829	86.39852	78.54464
Region & SAD_CON & DP	41.4766	82.24351	42.23272	83.99057	82.05863

Improvement step has no benefit at occluded region for SAD_MW_CON & DP and Region & DP algorithms of Venus image pair but at other regions of all algorithms, improvement has satisfactory results which can be seen in Table 5.9.

By the help of the improvement step, percentage of bad matching pixels of the fusion algorithms are decreased significantly. In order to compare the

performances of all of the algorithms, results for all data sets (Tsukuba, Venus and Sawtooth), are listed in Table 5.10. For comparison, *SAD_MW_CON* and *SAD_CON* are considered as correlation-based algorithms since they have better performances than *SAD* and *SAD_MW* and they can also find occluded pixels. Similarly, *Region_based_with_occlusion* algorithm is considered only rather than *Region_based* algorithm because occluded pixels are also detected by this algorithm. Improved fusion algorithms are used instead of fusion algorithms because of the satisfactory results presented in Table 5.9.

Table 5.10 The percentage of bad matching pixels of all algorithms are given all together for five quality measures for each data set.

TSUKUBA IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
SAD_CON	55.8 ⁽¹¹⁾	16.5 ⁽¹⁰⁾	34.0 ⁽¹⁰⁾	23.7 ⁽¹⁰⁾	11.3 ⁽¹⁰⁾
SAD_MW_CON	35.0 ⁽¹⁾	20.2 ⁽¹¹⁾	25.2 ⁽⁶⁾	31.0 ⁽¹¹⁾	12.7 ⁽¹¹⁾
DP	51.1 ⁽⁸⁾	8.1 ⁽⁸⁾	20.1 ⁽²⁾	7.0 ⁽⁹⁾	8.9 ⁽⁸⁾
Region	54.1 ⁽¹⁰⁾	6.2 ⁽⁴⁾	15.4 ⁽¹⁾	3.3 ⁽²⁾	8.2 ⁽⁶⁾
SAD_MW_CON & DP	41.6 ⁽⁴⁾	5.5 ⁽²⁾	22.89 ⁽³⁾	5.33 ⁽⁶⁾	5.6 ⁽¹⁾
Region & SAD_MW_CON	38.6 ⁽²⁾	5.6 ⁽³⁾	23.3 ⁽⁵⁾	4.1 ⁽³⁾	6.7 ⁽³⁾
Region & DP	47.4 ⁽⁵⁾	5.1 ⁽¹⁾	22.92 ⁽⁴⁾	3.2 ⁽¹⁾	6.5 ⁽²⁾
Region & SAD_MW_CON & DP	41.2 ⁽³⁾	6.7 ⁽⁵⁾	28.8 ⁽⁷⁾	5.27 ⁽⁵⁾	7.72 ⁽⁵⁾
SAD_CON & DP	53.0 ⁽⁹⁾	7.04 ⁽⁷⁾	30.70 ⁽⁸⁾	6.11 ⁽⁷⁾	7.70 ⁽⁴⁾
Region & SAD_CON	50.1 ⁽⁷⁾	6.98 ⁽⁶⁾	30.71 ⁽⁹⁾	4.7 ⁽⁴⁾	8.6 ⁽⁷⁾
Region & SAD_CON & DP	47.5 ⁽⁶⁾	8.2 ⁽⁹⁾	36.3 ⁽¹¹⁾	6.13 ⁽⁸⁾	9.6 ⁽⁹⁾
VENUS IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt

Table 5.10 Continued.

SAD_CON	76.5 ⁽⁷⁾	12.8 ⁽⁹⁾	40.0 ⁽¹⁰⁾	21.6 ⁽¹⁰⁾	8.4 ⁽⁸⁾
SAD_MW_CON	54.9 ⁽¹⁾	18.1 ⁽¹¹⁾	28.8 ⁽¹⁾	31.7 ⁽¹¹⁾	11.2 ⁽¹⁰⁾
DP	57.0 ⁽²⁾	10.6 ⁽⁸⁾	30.8 ⁽⁴⁾	10.59 ⁽⁷⁾	10.6 ⁽⁹⁾
Region	66.7 ⁽³⁾	17.9 ⁽¹⁰⁾	30.1 ⁽³⁾	20.6 ⁽⁹⁾	16.5 ⁽¹¹⁾
SAD_MW_CON & DP	82.0 ⁽¹⁰⁾	3.4 ⁽¹⁾	29.4 ⁽²⁾	3.7 ⁽¹⁾	3.3 ⁽¹⁾
Region & SAD_MW_CON	76.9 ⁽⁹⁾	7.4 ⁽⁶⁾	31.5 ⁽⁵⁾	10.57 ⁽⁶⁾	5.8 ⁽⁶⁾
Region & DP	88.9 ⁽¹¹⁾	8.4 ⁽⁷⁾	31.8 ⁽⁶⁾	11.4 ⁽⁸⁾	6.9 ⁽⁷⁾
Region & SAD_MW_CON & DP	74.86 ⁽⁵⁾	6.6 ⁽⁵⁾	36.9 ⁽⁸⁾	9.7 ⁽⁵⁾	5.1 ⁽³⁾
SAD_CON & DP	76.8 ⁽⁸⁾	3.8 ⁽²⁾	36.1 ⁽⁷⁾	4.0 ⁽²⁾	3.6 ⁽²⁾
Region & SAD_CON	74.99 ⁽⁶⁾	6.538 ⁽³⁾	39.6 ⁽⁹⁾	9.05 ⁽³⁾	5.27 ⁽⁵⁾
Region & SAD_CON & DP	69.7 ⁽⁴⁾	6.544 ⁽⁴⁾	41.4 ⁽¹¹⁾	9.10 ⁽⁴⁾	5.25 ⁽⁴⁾
SAWTOOTH IMAGE PAIR					
ALGORITHM	Bo	Bo*	Bd	Bt*	Bt
SAD_CON	46.9 ⁽⁵⁾	7.4 ⁽¹⁰⁾	33.4 ⁽¹⁰⁾	5.8 ⁽⁹⁾	7.5 ⁽¹⁰⁾
SAD_MW_CON	29.0 ⁽¹⁾	7.0 ⁽⁹⁾	23.1 ⁽⁴⁾	6.3 ⁽¹⁰⁾	7.0 ⁽⁹⁾
DP	57.7 ⁽⁹⁾	9.1 ⁽¹¹⁾	29.2 ⁽⁷⁾	5.7 ⁽⁸⁾	9.4 ⁽¹¹⁾
Region	53.3 ⁽⁷⁾	6.4 ⁽⁸⁾	14.6 ⁽¹⁾	10.5 ⁽¹¹⁾	6.0 ⁽⁸⁾
SAD_MW_CON & DP	58.9 ⁽¹¹⁾	4.4 ⁽³⁾	23.8 ⁽⁵⁾	3.2 ⁽²⁾	4.5 ⁽³⁾
Region & SAD_MW_CON	37.5 ⁽²⁾	3.7 ⁽¹⁾	22.2 ⁽³⁾	2.5 ⁽¹⁾	3.8 ⁽¹⁾
Region & DP	55.3 ⁽⁸⁾	4.2 ⁽²⁾	21.4 ⁽²⁾	4.7 ⁽⁶⁾	4.2 ⁽²⁾
Region & SAD_MW_CON & DP	46.2 ⁽⁴⁾	4.69 ⁽⁴⁾	27.0 ⁽⁶⁾	4.0 ⁽⁴⁾	4.76 ⁽⁴⁾
SAD_CON & DP	58.1 ⁽¹⁰⁾	5.2 ⁽⁶⁾	30.7 ⁽⁸⁾	4.3 ⁽⁵⁾	5.3 ⁽⁶⁾
Region & SAD_CON	45.5 ⁽³⁾	4.70 ⁽⁵⁾	31.2 ⁽⁹⁾	3.7 ⁽³⁾	4.81 ⁽⁵⁾
Region & SAD_CON & DP	47.5 ⁽⁶⁾	5.5 ⁽⁷⁾	34.3 ⁽¹¹⁾	5.1 ⁽⁷⁾	5.5 ⁽⁷⁾

According to Table 5.10, in order to find the best algorithm for each of the data set, the average rank value of all regions can be computed for the specified algorithm by using Eq. (5.3). Table 5.11 presents the average rank value of all algorithms for each data set.

Table 5.11 The average rank value of all algorithms for each data set.

AVERAGE RANK VALUES			
ALGORITHM	TSUKUBA	VENUS	SAWTOOTH
SAD_CON	10.2	8.8	8.8
SAD_MW_CON	8	6.8	6.6
DP	7	6	9.2
Region	4.6	7.2	7
SAD_MW_CON & DP	3.2	3	4.8
Region & SAD_MW_CON	3.2	6.4	1.6
Region & DP	2.6	7.8	4
Region & SAD_MW_CON & DP	5	5.2	4.4
SAD_CON & DP	7	4.2	7
Region & SAD_CON	6.4	5.2	5
Region & SAD_CON & DP	8.6	5.4	7.6

According to Table 5.11, *Region & DP* algorithm is the best one for Tsukuba, *SAD_MW_CON & DP* algorithm for Venus and *Region & SAD_MW_CON* algorithm for Sawtooth.

5.3 DISPARITY SEGMENTATION

Disparity segmentation block is explained in Chapter 4. Here, first of all, the disadvantage of applying mean shift segmentation to the whole disparity map is explained by using ground truth disparity map of Tsukuba which is shown in Figure 5.7.

Figure 5.7 Applying mean shift segmentation to the whole image of ground truth for Tsukuba. (a) Original image, (b) Ground truth, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3, (f) Segmented region 4, (g) Segmented region 5, (h) Segmented region 6.

According to Figure 5.7, it can be seen that mean shift segmentation has failed in the depth regions which are back to back and closer to each other. For example, the objects on the table and the head can not be segmented correctly which is shown in Figure 5.7 (d). For solving this problem, first the depth regions are found and then the mean shift segmentation process is applied to these regions. That's why the technique shown in Figure 4.4 is used in the rest of this section for observing the results of segmentation process.

For observing the results of disparity segmentation block, first the segmentation results of the ground truth disparity maps are going to be tested. Then the best disparity map for each of the data sets (*Region & DP* for Tsukuba, *SAD_MW_CON & DP* for Venus and *Region & SAD_MW_CON* for Sawtooth) are going to be tested. Parameters of the algorithms which are defined in Chapter 4 are set as shown in Table 5.12.

Table 5.12 Parameters of stereo analysis and disparity segmentation for data sets.

Parameters	TSUKUBA	VENUS	SAWTOOTH
<i>Max_Disparity</i>	20	20	20
<i>Region_Threshold</i>	Scale factor / 2	Scale factor / 2	Scale factor / 2
<i>Min_Seg_Area</i>	5000	15000	15000

Min_Seg_Area is much larger for Venus and Sawtooth than for Tsukuba because Tsukuba has a lot of distinct objects and they have smaller size than the objects in Venus and Sawtooth.

Segmentation of ground truth disparity maps is shown in Figure 5.8, Figure 5.9 and Figure 5.10. Segmentation of the disparity maps of the best resulting

improved fusion algorithms for data sets is shown in Figure 5.11, Figure 5.12 and Figure 5.13.

Figure 5.8 Segmentation of ground truth for Tsukuba. (a) Original image, (b) Ground truth, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3, (f) Segmented region 4, (g) Segmented region 5.

Figure 5.9 Segmentation of ground truth for Venus. (a) Original image, (b) Ground truth, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3.

Figure 5.10 Segmentation of ground truth for Sawtooth. (a) Original image, (b) Ground truth, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3.

Figure 5.11 Disparity segmentation for Tsukuba. (a) Original image, (b) Disparity map of Region & DP, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3.

Figure 5.12 Disparity segmentation for Venus. (a) Original image, (b) Disparity map of SAD_MW_CON & DP, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3.

Figure 5.13 Disparity segmentation for Sawtooth. (a) Original image, (b) Disparity map of Region & SAD_MW_CON, (c) Segmented region 1, (d) Segmented region 2, (e) Segmented region 3.

The segmented regions are specified as white and each is shown in a different figure. Segmentation of ground truth disparity maps shown in Figure 5.8, Figure 5.9 and Figure 5.10 indicates that the objects at different depths can be extracted and foreground-background segmentation can be done by applying segmentation to disparity map of the scene. According to Figure 5.8, it can be seen that the segmentation process has satisfactory results when the technique shown in Figure 4.4 is used. The objects which are closer to each other can be segmented successfully. As a result, the drawbacks seen in Figure 5.7 can be handled by analyzing the depth regions before segmentation process as shown in Figure 5.8.

In Figure 5.11, Figure 5.12 and Figure 5.13, the disparity maps of the best resulting improved fusion algorithms for data sets are used. Although it can be seen

that the segmentation of disparity maps is not as efficient as the ones obtained by the ground truth maps due to some false disparities, the results are still satisfactory.

In Figure 5.11, the two depths which are the closest to camera could be segmented and others are specified as background for Tsukuba. There are some false matches in the disparity map and the unnecessary smoothing effect of improvement step exists at the depth regions which are far away from camera. Therefore these problems affect the segmentation process.

In Figure 5.12 and Figure 5.13, the segmentation process gets harder and has some mistakes because disparity maps have some false matches and occluded regions at the boundaries of objects in Venus and Sawtooth data sets.

5.4 STEREO ANALYSIS AND DISPARITY SEGMENTATION OF REAL-TIME STEREO PAIRS

The data sets that we have used in this thesis are constructed manually by paying attention to lighting of the environment and the placement of objects. In order to test the performance of stereo algorithms in real life and applications, some real stereo images are also tested. In this section, different real life stereo pairs are tested and the results of the previously tested stereo algorithms are presented.

Firstly, the foreground-background segmentation procedure is going to be tested. For this reason, three different stereo pairs are used. These are shown in section 5.4.1, 5.4.2 and 5.4.3.

5.4.1 TEST STEREO PAIR # 1

The first stereo pair shown in Figure 5.14 is taken with a single camera by shooting one picture and then sliding the camera on the horizontal line and shooting once more. This is the fundamental and simple test image. The depth of background is kept as constant in order to determine the performance of stereo analysis and

disparity segmentation easily. Only one object (Object 1) is placed in the foreground and stereo analysis and disparity segmentation are implemented using this two disparity layered settings. The parameters are set as shown in Table 5.13 and the results are shown in Figure 5.15.

Figure 5.14 Test stereo pair # 1.

Table 5.13 Parameters of stereo analysis and disparity segmentation for test stereo pair # 1.

Parameters	TEST STEREO PAIR # 1
<i>Max_Disparity</i>	40
<i>Region_Threshold</i>	Scale factor / 2
<i>Min_Seg_Area</i>	5000

Figure 5.15 Stereo analysis and disparity segmentation of test stereo pair # 1. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

The disparity maps have some occluded regions as shown in Figure 5.15. Occluded regions are shown as black and generally exist near the boundary of objects. Although these regions make segmentation process difficult, the segmentation block used in this thesis have satisfactory results as shown in Figure 5.15.

5.4.2 TEST STEREO PAIR # 2

The second stereo pair shown in Figure 5.16 is taken from [45] which uses rectified stereo sequences acquired with a digital stereo camera. This is also a good test image for observing the performance of foreground-background segmentation. Object 1 is strictly far away from the background and depth of background is not constant. The parameters are set as shown in Table 5.14 and the results are shown in Figure 5.17.

Figure 5.16 Test stereo pair # 2 [45].

Table 5.14 Parameters of stereo analysis and disparity segmentation for test stereo pair # 2.

Parameters	TEST STEREO PAIR # 2
<i>Max_Disparity</i>	25
<i>Region_Threshold</i>	Scale factor / 2
<i>Min_Seg_Area</i>	5000

Figure 5.17 Stereo analysis and disparity segmentation of test stereo pair # 2. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

Although the depth of background is not constant, object 1 is segmented successfully by the segmentation block shown in Figure 5.17. Analyzing the depth regions facilitates the segmentation process in this test stereo pair. If the mean shift segmentation is applied to the disparity maps without finding depth regions, background can not be segmented as a whole.

5.4.3 TEST STEREO PAIR # 3

The third one shown in Figure 5.18 is the stereo pair obtained at MSRC (Microsoft Research Cambridge) [46]. This is similar with test stereo pair # 2. The parameters are set as shown in Table 5.15 and the results are shown in Figure 5.19.

Figure 5.18 Test stereo pair # 3 [46].

Table 5.15 Parameters of stereo analysis and disparity segmentation for test stereo pair # 3.

Parameters	TEST STEREO PAIR # 3
<i>Max_Disparity</i>	35
<i>Region_Threshold</i>	Scale factor / 2
<i>Min_Seg_Area</i>	5000

Figure 5.19 Stereo analysis and disparity segmentation of test stereo pair # 3. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

Foreground-background segmentation is successfully done as shown in Figures 5.15, 5.17 and 5.19. The object in the foreground is detected and extracted by segmenting the disparity maps. This is an easy process because one object is placed in the foreground. Parameters are set as default value because these images are not complex and do not have more than one object to be extracted at different depth layers. The performance of stereo and segmentation algorithms should be tested when more than one object can be placed at different depths. For this reason, other three different stereo pairs are used in sections 5.4.4, 5.4.5 and 5.4.6.

5.4.4 TEST STEREO PAIR # 4

The environment of test stereo pair # 1 has been changed and two objects have been placed at different depths as shown in Figure 5.20. Object 1 is more closer to camera than object 2 but both of them are far away from constant depth background. The parameters are set as shown in Table 5.16 and the results are shown in Figure 5.21.

Figure 5.20 Test stereo pair # 4.

Table 5.16 Parameters of stereo analysis and disparity segmentation for test stereo pair # 4.

Parameters	TEST STEREO PAIR # 4
<i>Max_Disparity</i>	30
<i>Region_Threshold</i>	Scale factor / 2.5
<i>Min_Seg_Area</i>	5000

Figure 5.21 Stereo analysis and disparity segmentation of test stereo pair # 4. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

As shown in Figure 5.21, object 1 and object 2 are segmented and extracted from background. The exact shape of the objects can not be formed in segmentation because of the quality of disparity maps. Stereo algorithms may have some false disparity values and unnecessary smoothing effect of improvement step may exist in some places in the disparity map. Therefore these situations affect the result of the segmentation process.

5.4.5 TEST STEREO PAIR # 5

This test stereo pair shown in Figure 5.22 is the same with test stereo pair # 4 but in this test environment, the difference between the depths of objects is larger

than the objects' of test stereo pair # 4. The parameters are set as shown in Table 5.17 and the results are shown in Figure 5.23.

Figure 5.22 Test stereo pair # 5.

Table 5.17 Parameters of stereo analysis and disparity segmentation for test stereo pair # 5.

Parameters	TEST STEREO PAIR # 5
<i>Max_Disparity</i>	25
<i>Region_Threshold</i>	Scale factor / 2
<i>Min_Seg_Area</i>	5000

Figure 5.23 Stereo analysis and disparity segmentation of test stereo pair # 5. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

The disparities of objects are estimated correctly which is shown in Figure 5.23. Object 1 and object 2 are extracted from the background successfully but the exact boundaries of objects can not be constructed because some false disparities are calculated near the object boundaries because of the smoothing effect of improvement step.

5.4.6 TEST STEREO PAIR # 6

Also new stereo pair which has been taken from stereo vision based robot system, is used as shown in Figure 5.24. In this test environment, object 1 and object 2 are much far away from each other and the background. The parameters are set as shown in Table 5.18 and the results are shown in Figure 5.25.

Figure 5.24 Test stereo pair # 6.

Table 5.18 Parameters of stereo analysis and disparity segmentation for test stereo pair # 6.

Parameters	TEST STEREO PAIR # 6
<i>Max_Disparity</i>	30
<i>Region_Threshold</i>	Scale factor / 1.5
<i>Min_Seg_Area</i>	5000

Figure 5.25 Stereo analysis and disparity segmentation of test stereo pair # 6. (a) Region & DP (b) SAD_MW_CON & DP, (c) Region & SAD_MW_CON.

Only region-based stereo algorithm can handle the problem of large textureless region which is explained in section 5.2.3. Object 1 has large textureless region and has some false disparities in the disparity map of SAD_MW_CON & DP because of the lack of region-based algorithm so object 1 is not segmented successfully as shown in Figure 5.25 (b). Unnecessary smoothing effect of improvement step also exist in all disparity maps in Figure 5.25 so the exact shape of objects can not be extracted by segmentation process.

The *Region_Threshold* parameter also affects the segmentation process. It should be chosen according to the structure of the environment which means that the placement of objects is very important for *Region_Threshold*. When the difference between the depth of objects increases, *Region_Threshold* should be increased.

5.5 STEREO ANALYSIS OF SOME STEREO VIDEO SEQUENCES

So far, stereo algorithms have been tested by using real time stereo images and stereo data sets. In this section, the frames of a stereo video are tried to be analyzed. Therefore a stereo vision based robot system can be used in order to test the stereo analysis. The frames of stereo video are processed and the disparity map of the environment is modeled for stereo video#1 and video#2 as shown in Figure 5.26 and Figure 5.27. The SAD_MW_CON & DP algorithm is used in these experiments.

Figure 5.26 Outputs of SAD_MW_CON & DP algorithm for frames of stereo video#1. (a) Frame 1, (b) Frame 2, (c) Frame 3, (d) Frame 4, (e) Frame 5, (f) Frame 6, (g) Frame 7, (h) Frame 8, (i) Frame 9, (j) Frame 10.

Figure 5.27 Outputs of SAD_MW_CON & DP algorithm for frames of stereo video#2. (a) Frame 1, (b) Frame 2, (c) Frame 3, (d) Frame 4, (e) Frame 5.

In Figure 5.26 and Figure 5.27, stereo cameras are constant and the depth of men varies frame by frame. In spite of the false disparity values and unnecessary smoothing effect of improvement step, disparity maps can be constructed correctly. As a result of Figure 5.26 and Figure 5.27, the localization of a robot, the 3D reconstruction of an environment and etc. can be considered as an application area of the methodology used in this study.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this thesis, disparity information of pixels is obtained by realizing stereo analysis of the environment. Foreground-background is segmented and objects in different depths are tried to be extracted by using disparity segmentation. Proposed method is based on developing the robust and reliable stereo analysis for nearly real-time applications and detecting objects by using reliable segmentation procedure.

In this study, first, suitable stereo algorithms (correlation-based, dynamic programming and region-based) for real-time applications are selected and their performances are compared by using some common stereo data sets. Moreover, the effects of the parameters are analyzed for each method separately by using quality measures. For example, in SAD correlation-based algorithms, selecting an appropriate window size is a basic problem. If the window size is too large, object borders are blurred and small details or objects are removed. If the window size is too small, it provides high performance at the object borders but it increases the influence of noise and decreases the correct matches. Therefore the suitable window size (between 9x9 and 15x15) is determined by the experimental tests in this thesis. Furthermore, the effect of variance of camera sensor noise in the data sets and its optimal value are presented in dynamic programming experiment. The streak problem of dynamic programming can not be solved because of high computational complexity. On the other hand, in region-based algorithms, the size of the segmented

regions is important and if the image has large textureless regions, the size of the segments must be large enough for handling false matches.

The fusion of the stereo algorithms is realized successfully in this thesis. It combines the disparity maps of stereo algorithms and accepts the same disparity values in all algorithms as true and others as unknown. Fusion generates more reliable disparity map but there can be some pixels which are not assigned disparity values after fusion process. These unknown disparities are tried to be estimated by using improvement step. Improvement is a disparity refinement procedure that uses continuity constraint for estimating unknown disparities. There are some disadvantages of fusion and improvement steps. For example, fusion process has high computational complexity and improvement step has unnecessary smoothing effects in some regions (especially borders of objects). In spite of all disadvantages, the performance of fusion and improvement steps are analyzed and compared with the stereo algorithms and it has been seen that the performance of stereo analysis gets much better after these steps.

Another important aspect of this study is to segment foreground-background and objects in different depths by using the output of stereo analysis i.e. the disparity map. First, the depth regions of disparity map are tried to be estimated by analyzing the pixels of disparity map according to some parameters which are related to the structure of the scene. Then, mean shift segmentation is applied to specified regions and objects are extracted roughly. Disparity map have some false matches inevitably and improvement step have some unnecessary smoothing effect in some places in the disparity map. So, the exact shape of objects can not be found after segmentation. That's why the quality of disparity map is so important and critical for segmentation process. Also to extract objects correctly, they should be far away from background and each other and there should not be large occluded parts in the image.

Consequently, the results of the implemented algorithms are encouraging for both data sets and test stereo pairs in this study.

6.2 FUTURE WORK

The algorithms used in this thesis are implemented in Matlab that's why they are not so fast as working in real-time applications. A first step forward from this work is to reimplement the algorithms in a high level programming language like C therefore study comes closer to real-time applications completely. In this situation, other global stereo methods can be used without avoiding their complexity because global methods are more efficient than others according to the performance in all regions of stereo vision like occluded, discontinuity, textureless, etc. So, the performance of stereo analysis can be efficiently increased and the performance of segmentation can be more better.

The stereo algorithms are implemented according to rectified stereo images in this thesis that's why they can only find matching points on the horizontal epipolar lines. They use geometric constraint exactly that matching points are searched in only one direction. These algorithms can be modified to work with unrectified stereo images by searching the matching points on not only horizontal epipolar lines but also in a specified region. This situation may result in a problem about computational complexity in Matlab but may not be in high level programming language.

The last step is to develop the segmentation process in order to increase its performance and reliability. Hybrid technique can be used which means color segmentation can be applied to original stereo image and results of color segmentation can be combined with the results of disparity map segmentation so the exact borders and the exact shape of the objects can be determined. In this situation, the application area of this study is highly increased.

REFERENCES

- [1] Canny J., “A Computational Approach To Edge Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.
- [2] Harris C., Stephens M., “A combined corner and edge detector”, In Alvey Vision Conference, pp. 147-151, 1988.
- [3] Wang H., Brady M., “Real-time corner detection algorithm for motion estimation”, Image Vision Comput. 13(9): 695-703, 1995.
- [4] Laganier R., “A Morphological Operator for Corner Detection”, Pattern Recognition, pp. 1643-1652, 1998.
- [5] Lee Y.S., Koo H.S., Jeong C.S., “A straight line detection using principal component analysis”, Pattern Recognition Letters 27(14): 1744-1754, 2006.
- [6] Lopez A., Pla F., “A minimization approach for 3D recovery in region-based stereo vision”, 7th International Conference on Image Processing and its Applications, IEE, pp. 47-51, 1999.
- [7] Cohen L., Vinet L., Sander P.T., Gagalowicz A., “Hierarchical region based stereo matching”, Computer Vision and Pattern Recognition, Proceedings CVPR '89., IEEE Computer Society Conference, pp. 416-421, 1989.
- [8] <http://www-prima.imag.fr/perso/Hall/Courses/FAI05/Session11.ppt>, last accessed date: 23.07.2008.
- [9] Lew M. S., Huang T. S., Wong K., “Learning and Feature Selection in Stereo Matching”, Pattern Analysis and Machine Intelligence, vol. 16, no. 19, pp. 869-881, September 1994.

- [10] Wang J. H., Hsieh R. F., and Chiu H. C., “A progressive constraint search approach for disparity matching in stereo vision”, Proceedings of the National Science Council – Part A, vol. 23, no. 6, pp. 789-798, June 1999.
- [11] <http://www.cs.bris.ac.uk/Teaching/Resources/COMS30121/slides3D/lec2P.pdf>, last accessed date: 23.07.2008.
- [12] Loop C. and Zhang Z., “Computing Rectifying Homographies for Stereo Vision”, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 125-131, June 23-25, 1999, Fort Collins, Colorado, USA.
- [13] Koschan A., “What is New in Computational Stereo Since 1989: A Survey of Current Stereo Papers”, Technical Report 93-22, Technical Univ. of Berlin, 1993.
- [14] Kuhl A., “Comparison of Stereo Matching Algorithms for Mobile Robots” The University of Western Australia, October 2004 – February 2005.
- [15] http://www.diss.fu-berlin.de/diss/servlets/MCRFileNodeServlet/FUDISS_derivate_000000002505/11_Chapter11.pdf, last accessed date: 23.07.2008.
- [16] Scharstein D., Szeliski R., “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, Intl. J. Comp. Vis., 47(1): 7–42, 2002.
- [17] Ogale A. S., Aloimonos Y., “Shape and the stereo correspondence problem”, International Journal of Computer Vision, vol. 65, Issue 3, pp. 147-162, December 2005.
- [18] Okutomi M., Kanade T., “A multiple baseline stereo”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 4, pp. 353-363, April 1993.

- [19] Fusiello A., Roberto V., Trucco E., “Efficient stereo with multiple windowing”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 858-863, June 1997.
- [20] Kanade T., Okutomi M., “A stereo matching algorithm with an adaptive window: theory and experiment”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, no. 9, pp. 920-932, 1994.
- [21] Cox I. J., Hingorani S. L., Rao S. B., Maggs B. M., “A maximum likelihood stereo algorithm”, Computer Vision and Image Understanding, 63(3):542–567, May 1996.
- [22] Sun J., Zheng N. N., Shum H. Y., "Stereo Matching Using Belief Propagation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, Jul., 2003.
- [23] Roy S., Cox I., “A maximum-flow formulation of the n-camera stereo correspondence problem”, Proc. of Int. Conf. on Computer Vision, pp. 492-499, 1998.
- [24] Boykov Y., Veksler O., Zabih R., “Fast approximate energy minimization via graph cuts”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222–1239, November 2001.
- [25] Omer I., Werman M., “Image Specific Feature Similarities”, School of Computer Science, The Hebrew University of Jerusalem, ECCV06 (II: 321-333).
- [26] Liao T., “Image Segmentation – Hybrid Method Combining Clustering and Region Merging”, Bachelor of Computer Science with Honours Thesis, Monash University, November 2003.

- [27] Peterson J. D., “Clustering overview”, <http://www.cs.ndsu.nodak.edu/~jasonpet/CSCI779/Clustering.pdf>, last accessed date: 24.07.2008.
- [28] Ersoy O., “Image Segmentation with Improved Region Modelling” Master of Science Thesis, METU 2004.
- [29] decsai.ugr.es/mia/complementario/t1/book3draft/clustering.pdf, last accessed date: 24.07.2008.
- [30] Wu Y. “Segmentation: Clustering, Graph Cut and EM”, ECE432 – Advanced Computer Vision Notes Series 5.
- [31] Lecture notes of Bryan S. Morse, “Segmentation (Region based)”, Brigham Young University, 3 March 2000.
homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/region.pdf, last accessed date: 24.07.2008.
- [32] Pujol X. M., “Image Segmentation Integrating Colour, Texture and Boundary Information”, PhD Thesis, Universitat de Girona, December 2002.
- [33] Çığla C., “Dense Depth Map Estimation For Object Segmentation In Multi-View Video”, Master of Science Thesis, METU July 2007.
- [34] Çığla C., Akpınar S., “Stereo for image based rendering using image over-segmentation”.
- [35] Demoulin C., Droogenbroeck M. V., “Disparity map determination by multiple adaptive windows”, In ProRISC/IEEE Workshop on Circuit, Systems and Signal Processing, Veldhoven, The Netherlands, pp. 615-618, November 2005.

- [36] Stefano L., Marchionni M., Mattoccia S., "A fast area-based stereo matching algorithm", *Image Vision Computing*, vol. 22, issue. 12, pp. 983-1005, 2004.
- [37] Hirschmüller H., Innocent P. R., Garibaldi J., "Real-time correlation-based stereo vision with reduced border errors", *International Journal of Computer Vision*, 47(1/2/3):229-246, 2002.
- [38] Veksler O., "Fast variable window for stereo correspondence using integral images", In *CVPR03*, pages I: 556–561, 2003.
- [39] Wagner D. B., "Power Programming: Dynamic Programming", vol. 5, Issue 4, Fall 1995.
<http://www.mathematica-journal.com/issue/v5i4/columns/wagner/42-51wagner.mj.pdf>, last accessed date: 24.07.2008.
- [40] <http://www.aw-bc.com/info/kleinberg/assets/downloads/ch6.pdf>, last accessed date: 24.07.2008.
- [41] Prof. Ramachandran V., "Dynamic Programming", The University of Texas at Austin, Department of Computer Sciences, Spring 2006.
<http://www.cs.utexas.edu/~vlr/s06.357/notes/lec8.pdf>, last accessed date: 24.07.2008.
- [42] Prof. Leiserson C. E., "Lectures Notes – Introduction to Algorithms : Dynamic Programming", 7 November 2005.
<http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-046JFall-2005/6939E270-F2FA-4C0E-921C-BBA7886B9573/0/lec15.pdf>, last accessed date: 24.07.2008.
- [43] Sabihuddin S. and MacLean W. J., "Maximum-Likelihood Stereo Correspondence using Field Programmable Gate Arrays", In *Proceedings of the 5th*

International Conference on Computer Vision Systems (ICVS2007), Bielefeld, Germany, 2007.

[44] Comaniciu D., Meer P., “Mean Shift : A robust approach toward feature space analysis”, IEEE Transactions on pattern analysis and machine intelligence vol. 24, no. 5, May 2002.

[45] <http://www.vision.deis.unibo.it/smatt/stereo.htm>, last accessed date: 09.08.2008.

[46] <http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm>, last accessed date: 09.08.2008.

[47] http://www.csc.kth.se/utbildning/kth/kurser/DD2422/bildat08/08_lecture12_6.pdf, last accessed date: 19.08.2008.

[48] <http://vision.middlebury.edu/stereo>, last accessed date: 05.10.2008.

[49] Forstmann S., Ohya J., Kanou Y., Schmitt A., Thuring S., “Real-Time Stereo by using Dynamic Programming”, In Proc. of CVPR Workshop on Real-time 3D Sensors and Their Use, 2004.

APPENDIX

STEREO DATA SETS

Tsukuba Stereo Pair :

Venus Stereo Pair :

Sawtooth Stereo Pair :