DENSE DEPTH MAP ESTIMATION FOR OBJECT SEGMENTATION IN MULTI-VIEW VIDEO

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

BY

CEVAHİR ÇIĞLA

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

JULY 2007

Approval of the Thesis

"DENSE DEPTH MAP ESTIMATION FOR OBJECT SEGMENTATION IN MULTI-VIEW VIDEO"

Submitted by **CEVAHİR ÇIĞLA** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering** by,

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

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** _____

Assoc. Prof. Dr. A. Aydın Alatan
Supervisor, **Electrical and Electronics Engineering, METU** _____

Examining Committee Members

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

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

Asst. Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering, METU _____

Asst. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Asst. Prof. Dr. Erhan Eren
Informatics Institute, METU _____

Date: **23.07.2007**

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 : Cevahir Çığla

Signature :

ABSTRACT

DENSE DEPTH MAP ESTIMATION FOR OBJECT SEGMENTATION IN MULTI-VIEW VIDEO

Çığla, Cevahir
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

July 2007, 138 Pages

In this thesis, novel approaches for dense depth field estimation and object segmentation from mono, stereo and multiple views are presented. In the first stage, a novel graph-theoretic color segmentation algorithm is proposed, in which the popular Normalized Cuts [6] segmentation algorithm is improved with some modifications on its graph structure. Segmentation is obtained by the recursive partitioning of the weighted graph. The simulation results for the comparison of the proposed segmentation scheme with some well-known segmentation methods, such as Recursive Shortest Spanning Tree [3] and Mean-Shift [4] and the conventional Normalized Cuts, show clear improvements over these traditional methods.

The proposed region-based approach is also utilized during the dense depth map estimation step, based on a novel modified plane- and angle-sweeping

strategy. In the proposed dense depth estimation technique, the whole scene is assumed to be region-wise planar and 3D models of these plane patches are estimated by a greedy-search algorithm that also considers visibility constraint. In order to refine the depth maps and relax the planarity assumption of the scene, at the final step, two refinement techniques that are based on region splitting and pixel-based optimization via Belief Propagation [32] are also applied.

Finally, the image segmentation algorithm is extended to object segmentation in multi-view video with the additional depth and optical flow information. Optical flow estimation is obtained via two different methods, KLT tracker and region-based block matching and the comparisons between these methods are performed. The experimental results indicate an improvement for the segmentation performance by the usage of depth and motion information.

Key words: Graph-theoretic image segmentation, dense depth map estimation, plane and angle sweeping, multi-view video object segmentation.

ÖZ

ÇOK GÖRÜNTÜLÜ VİDEODA NESNE BÖLÜTLEMESİ İÇİN SIK DERİNLİK HARİTASI KESTİRİMİ

Çığla, Cevahir
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. A. Aydın Alatan

Temmuz 2007, 138 sayfa

Bu tezde, sık derinlik haritası çıkarımı ile tek, stereo ve çoklu görüntüden nesne bölütlemesi problemleri için öne sürülen yeni yaklaşımlar sunulmaktadır. İlk kısımda, yaygın olarak kullanılan düzgülü kesik görüntü bölütleme algoritmasının çizge yapısı üzerinde yapılan değişikliklerle geliştirilmesi ile oluşturulan çizge tabanlı renk bölütlemesi algoritması önerilmektedir. Bölütleme çizgenin döngüsel olarak parçalara ayrılmasıyla elde edilir. Önerilen yöntemin bazı iyi bilinen bölütleme algoritmaları, döngülü en kısa kapsayan ağaç [3], ortalama kayma [4] ve klasik düzgülü kesik [6], ile karşılaştırılması amacıyla yapılan deneylerin sonuçları klasik yöntemler üzerindeki gelişmeleri açıkça göstermektedir.

Bölgesel tabanlı yaklaşım aynı zamanda yeni geliştirilmiş düzlem ve açı taramasına dayalı sık derinlik haritası kestirimi aşamasında da kullanılmaktadır. Önerilen sık derinlik haritası kestirimi yönteminde tüm sahnenin bölgesel olarak düzlemlerden oluştuğu varsayılmaktadır. Düzlemsel yamaların 3 boyuttaki

modelleri görünürlük kısıdı da kullanan fırsatçı bir arama algoritması ile kestirilmektedir. Son aşamada, derinlik haritalarını iyileştirmek ve sahnenin düzlemselliğini gevşetmek için bölge parçalama ve piksel tabanlı yargı yayılımına [32] dayalı iki farklı yöntem önerilmektedir.

Son olarak, görüntü bölütleme algoritması derinlik ve optik akış bilgilerinin eklenmesi ile çoklu görüntülü video nesne bölütlemesi amacıyla genişletilmektedir. Optik akış iki farklı yöntemle, "KLT izleme" ve bölge tabanlı blok eşleme, elde edilmektedir. Her iki yöntem de, önerilen bölütleme algoritmasındaki kullanılabilirlikleri açısından karşılaştırılmaktadır. Yapılan deneyler, renk bilgisine ilave olarak kullanılan derinlik ve optik akış bilgilerinin bölütleme performansını arttırdığını göstermektedir.

Anahtar Kelimeler: Çizge kuramlı görüntü bölütleme, sık derinlik haritası kestirimi, düzlem ve açı tarama, çoklu görüntülü video nesne bölütlemesi

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to my supervisor Assoc. Prof. Dr. Aydın Alatan for his guidance, suggestions and also for the great research environment he had provided.

I would like to thank my family and my love Hande for their understanding, support and patience throughout this work.

I would like to also express my thanks for their great friendship and assistance to Yoldaş Ataseven, Oytun Akman and Ahmet Saracoğlu. We were together for two invaluable years and I surely miss working with them.

I would also express my gratitude to Xenophon Zabulis for his brilliant ideas and suggestions throughout this work.

Finally, I would like to thank my friends in Multimedia Research Group for such a friendly research environment they had provided. I also want to thank Evren İmre and Alper Koz, I have learned much from our technical discussions, their suggestions and experiences.

This work is funded b*y EC IST 6th Framework 3DTV NoE* and partially funded by TÜBİTAK under Career Project 104E022.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Computer vision is a discipline with wide application areas and attracts many researchers due to its variety of innovative outcomes. The robot navigation systems, human motion modeling and its adaptation to animation technology, object segmentation and recognition, target detection, traffic surveillance, 3D data extraction and environment modeling are all active research areas under computer vision area. The recent advances in computer vision pioneered new technological devices and ideas that will affect future research directions. One of the most interesting and attractive advance is the 3D Television, *aka* 3DTV, which is supposed to be an important consumer electronic product in the near future.

The idea of 3DTV has canalized the researchers towards 3D data processing, involving extraction, coding, transmission, and visualization. As 3D data processing has been improved, the application area expanded to each problem in computer vision. Hence, especially in the last decade, the algorithms previously developed for 2D data have been improved and adapted to their 3D counterparts. However, the transition from 2D to 3D results in diverse problems and the solutions to these problems are obtained via different assumptions and methods which are generally related to 3D geometry.

The modeling of the 3D environment from 2D data and its understanding (analysis) are two of the fundamental and common problems for the conversion of

2D to 3D. The modeling part involves the extraction of 3D information of a scene, such as depth or surface characteristics from two or multiple 2D images. With the recent developments in the capture technology, multiple-video concept becomes popular, instead of the conventional 2D video. 3D modeling of the scene, which is extracted from the multi-view video, could be utilized for various purposes, such as multi-view video coding, image rendering from arbitrary camera positions, which are vital for 3D application area.

Another fundamental problem is the analysis of the scene which can be examined in object segmentation and recognition problems. The object segmentation is the grouping of pixels into meaningful classes depending on the image structure, such as color and depth. In 3D applications, segmentation is also important for coding and image based rendering, especially when combined with the depth or surface information.

1.1 Scope of the Thesis

The motivation behind this study is supported by two main propositions. The first proposition is the excessive requirement of 3D structure in multimedia applications and its importance for the next generation research areas. The second one is the application of the extracted 3D information for the segmentation of a scene and extension of the well-known traditional 2D segmentation methods to 3D. The research in this thesis is devoted to the extraction of 3D information from 2D data via dense depth map estimation and its application to image segmentation.

The dissertation can be analyzed in three main steps performed sequentially for the main purpose of the thesis. The first step is the development and evaluation of the performance of some image segmentation algorithms for 2D images. This step

provides a comparison between four different algorithms which utilize the color and intensity distribution among the images. In the next step, the estimation of the 3D structure of a scene from 2D images is performed. The structure or the geometry is modeled via a novel dense depth map estimation algorithm. The stereo matching and its multi-view extension are analyzed and the solution is proposed based on the planarity assumption of the scene. Finally, the 2D segmentation and the dense depth map estimation methods are combined in order to propose a solution to the 3D segmentation problem in video.

1.2 Outline of the Thesis

The thesis is composed of three main parts involving the steps mentioned previously.

Chapter 2 focuses on 2D segmentation methods and their comparison. First, a literature survey on color image segmentation algorithms is given and the methods are classified. Then, a modified version of a well-known global segmentation algorithm based on graph-cuts is introduced with its advantages over the previous version. Finally, the comparison of the proposed method with two popular methods, recursive shortest spanning tree and mean-shift, and normalized cuts method is achieved by the simulations on a variety of images.

In Chapter 3, a novel dense depth map estimation algorithm from fully calibrated stereo and multiple images is proposed. Initially, some background information on the camera geometry and epipolar geometry is given. In the next step, the literature review of the stereo and multi-view dense depth map estimation algorithms is presented with the fundamental assumptions. In the final step, a novel approach to region-based dense depth map estimation algorithms is introduced supported with the experimental results.

Chapter 4 is devoted to the utilization of the proposed 2D segmentation algorithm with the 3D information extracted via dense depth matching and the optical flow information. First, a brief literature review on video object segmentation is given. Then, the segmentation with only one image is extended to multi-images and multi-view video. The performance changes on the segmentation quality are analyzed by the simulations on multi-view video in the experimental results section.

Finally, Chapter 5 gives the summary of thesis and the conclusion on the proposed region-based dense depth estimation algorithm and its application to 3D segmentation. In addition, the future directions involving the enhancement of the proposed algorithm are described with some recommendations and remarks.

CHAPTER 2

COLOR SEGMENTATION

One of the most challenging problems in computer vision is color segmentation, which is also an important tool for image understanding. Segmentation is an image processing task that aims at partitioning an image into homogeneous regions in terms of the features of pixels extracted from the image [1]. It is also desired to obtain those regions to be semantically meaningful. Segmentation results are usually utilized in some high-level operations, such as recognition and representation. The most well-known application areas of segmentation are vision-guided autonomous robotics, medical diagnosis, segment-based video compression, and the analysis of remotely sensed images [1].

Although many different approaches to the segmentation problem have been proposed, the problem has not been (and still far from being) totally solved due to its complicated range space. The problem stems from the non-uniqueness of the segmentation such that even different people might partition a presented image in different ways. However, the uniqueness of the segmentation can be provided via considering some constraints and obtaining segmentations for particular and special purposes. The survey of the different segmentation algorithms in literature is given in [1] with an excellent taxonomy.

The segmentation methods can be classified into two main distinct groups, the histogram-based and pixel-based methods. In the histogram-based methods [2],

the neighborhood relations between pixels are not considered and the segmentation is performed only according to the color distribution observed by histograms, which is actually a global approach. Hence, the resulting regions may have disjoint pixels which is generally not a preferred case. However, these methods yield satisfactory results, especially under some controlled conditions, such as visual product defect control in industrial applications.

On the other hand, the pixel-based methods utilize the neighboring relations between pixels and divide images into connected regions. The pixel-based methods can also be classified into two sub-groups, the local and global methods. In the local methods [3] [4], the clusters are obtained by combining the pixels, the smallest element in an image, and the segmentation is achieved with a bottom-to-top approach. A traditional down-to-top approach is reversed in global methods [6] [7] [8], an image is considered as a "big picture" and the segmentation is achieved downward by splitting it into smaller regions. Both of these methods are widely used and have characteristic advantages and disadvantages. In [3] [4], the pixels belonging to the same object can be segmented separately, since only the local properties are considered during the clustering of the pixels. The perceptional properties are utilized in global methods [6] [7] [8]; and the clustering is performed by a similar mechanism as in human perception, from whole to the detail. Although the global methods have a very important and realistic assumption, the computation can be time consuming as the image size increases. The computation problem is usually tried to be solved by down sampling the images and performing a multi resolution approach [9]. However, the details of the images are usually lost during these operations. Moreover, the local methods perform faster and are more applicable to real time systems than the global methods. There are also hybrid methods [10] which utilize different segmentation algorithms in one algorithm and have characteristics between local and global.

In this work, the segmentation problem is examined by comparing four different methods. The color segmentation performances of Recursive Shortest Spanning Tree (RSST) [3], Mean-Shift [4] and Normalized Graph Cuts [6] are investigated and a modified version of [6] is introduced in order to increase the speed and the performance of graph-based segmentation algorithms. In the proposed method, the graph is constructed by similarities between sub-regions obtained via over-segmentation by local methods instead of between pixels, which causes the graph based approaches to be slower. Then the segmentation is performed by partitioning the graph and grouping the sub-regions into larger clusters.

This chapter is composed of six main sections; in the first part the definitions and mathematical expressions of graph theory are presented. In the second and third sections, as being local methods, RSST and Mean-Shift segmentation algorithms are explained. The following section is devoted to normalized graph cut algorithm, which is one of the most well-known graph-based techniques and has introduced a novel approach to the image segmentation problem. In a different section the modified version of normalized graph cuts method is proposed and discussed with its advantages over [6]. Finally, the comparison among four different algorithms is performed in the experimental results section with the resultant segmentations of different set of images.

2.1 Fundamental Definitions of Graph Theory

A graph is a set of *vertices* (nodes) and *links* (edges) in which, the vertices are connected to each other via links. The mathematical representation of a graph is given as $G = (V, E)$, where V is the vertices and E is the links between vertices. In Figure 2.1, an example of a graph is illustrated with seven nodes and their connections with each other. In a weighted graph, the links are weighted and the connection strength among vertices is modeled via corresponding weights. If all

the vertices are connected each other in a graph, then the graph is denoted as a *complete graph*. However, each vertex is not necessarily linked to every other vertex and such a graph is called *partial graph.* For some graphical structures, the direction of the link can be important for the weighting of the graph and the weights may change according to the direction, this kind of graphs are called as *directed weighted graphs*. If the link between vertices shows the similarity of the nodes hence the direction is not important, then the graph is an *undirected weighted graph*.

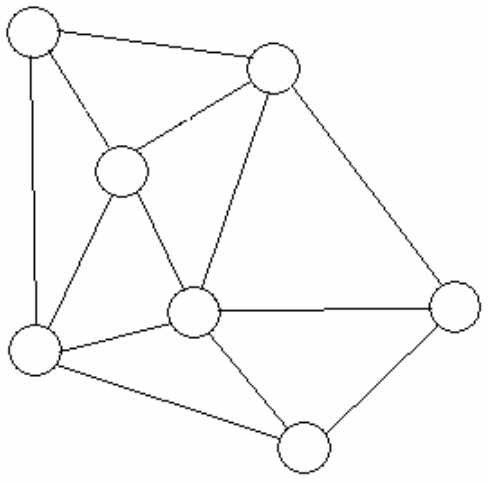


Figure 2.1: A simple graph with seven vertices

2.2 Segmentation with Recursive Shortest Spanning Tree

RSST [3] is a powerful segmentation method that is a hierarchical segmentation scheme yielding various scales of segmentation masks. In addition, RSST requires no initial segmentation input and parameters, and performs relatively fast. In the

hierarchical scheme, the algorithm evolves from coarse to fine and the segmentation can be obtained with the desired number of regions.

There are two functional blocks in the RSST algorithm, the *initialization stage* and the *linking process* as illustrated in Figure 2.2. In the initialization step, a weighted graph, whose nodes are the pixels and the links as a function of pixels intensity values, is constructed. Hence, the image is mapped onto a graph with number of vertices equal to the number of pixels in the image. In the next steps, the vertices correspond to regions with certain amount of pixels and link weights between the vertices are calculated via the mean intensity of the regions. The mean intensity of a region is defined as vertex weight (V_i), and the link weights ($LW(i,j)$) are evaluated by a function of V_i's and the sizes of the regions (N_i) which indicate the number of pixels in regions. Although, there could be different cost functions, the following linking cost function is preferred [3]:

$$LW(i,j) = \left|V_i - V_j\right| \frac{N_i N_j}{N_i + N_j} \tag{2.1}$$

After the construction of the graph, the links are sorted according to their link weights and the weakest link with the smallest cost value is detected. Then, the regions connected to each other through the weakest link are merged and the graph structure changes with a decrease in the number of the nodes. The vertex weight of the merged region is then updated and the link weights corresponding to all of the surrounding vertices are also revised with the new vertex weight of the merged region by (2.1). The merging process is illustrated in Figure 2.3. The number of the vertices is decreased by one with one linking operation involving sorting and merging successively. The linking process is iterated until the desired number of vertices which is actually the required number of regions is obtained. The hierarchical scheme of RSST can also be performed by saving the weakest links detected in linking process and utilizing the same order for different scales of the image [11].

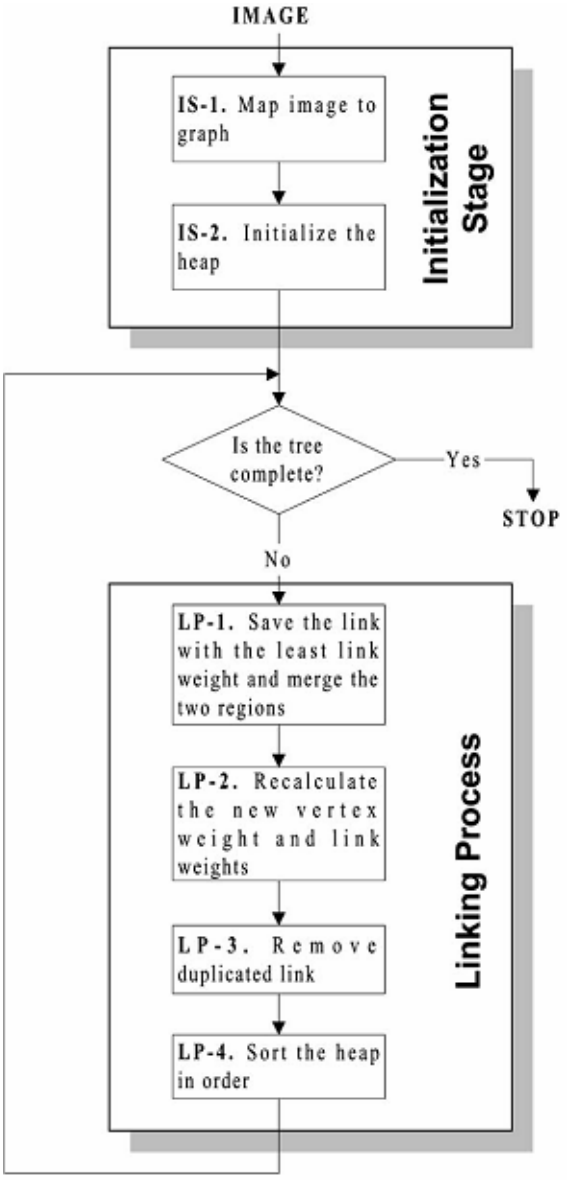


Figure 2.2: The flowchart of the RSST algorithm [3]

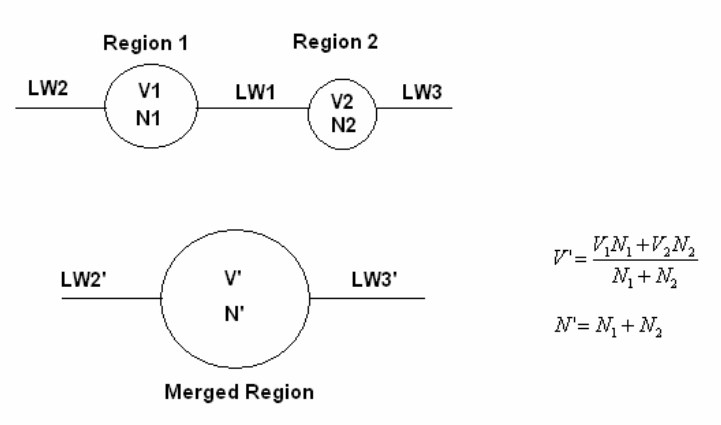


Figure 2.3: LW1 is the weakest link before merging and V', N' are the merged region properties.

2.3 Mean Shift Segmentation

Mean-Shift (MS) is an extremely versatile tool for feature space analysis and can provide reliable solutions for many vision tasks [4]. MS is widely used in many purposes due to its adaptable and excellent qualities. During the last few years, segmentation has been one of the most important application areas of MS with high quality results.

MS segmentation procedure is an extension of smoothing the observed data with kernels and labeling the pixels accordingly. The data, image in this case, is initially mapped to another color space so that the perceived color differences correspond to *Euclidean* distances in color space [4]. A *Euclidean* space for a color space is not guaranteed; however *L*u*v* and *L*a*b* spaces were designed to approximate uniform color spaces perceptually [4]. The *RGB* color space does not provide independent bases, and the new space *L*u*v* is constructed by the nonlinear dependency on *RGB*. *L* corresponds to the "*lightness*" or "*luminance*"

and the other coordinates define the "*chrominancy*". After the color space transformation, the data is filtered as follows:

1. For a pixel (x) in the image, evaluate the mean shift vector by (2.2)

$$f(x) = \frac{1}{k} \sum_{x_i \in S_h(x)} L(x_i).(x_i - x) \qquad (2.2)$$

   where $S_h(x)$ is the hyper sphere centered at x with radius h and k is the number of pixels within the sphere.
2. Then, assign the mean value to the corresponding pixel and shift the center of the hyper sphere to the new location. .

$$x' = f(x)$$

3. Evaluate the mean shift vector with the updated terms.
4. Iteratively perform the first three operations until $f(x)$ convergences to a number y.
5. After executing the first four operations for all of the pixels in the image, the filtered data is obtained which is composed of y's.

The mean shift operation and its convergence are illustrated with an example given in Figure 2.4. As observed in the example, the mean shift vector points towards the direction of the maximum increase in the density. The assignment of the convergent values to the pixels is performed after mean shift operation is employed for all of the pixels. Thus, the convergent mean values form the smoothed data.

After the smoothing, the clusters $\{C_p\}_{p=1...N}$ are determined from the set of $\{y_{ij}\}$'s by grouping the values which are closer than a pre-determined threshold [4]. Finally, for each pixel a labeling to the set of clusters is performed,

$L(x_{ij}) = \{p \mid y_{ij} \in C_p\}$. The steps of the algorithm are illustrated with an example in Figure 2.5, the small gradient changes are successfully handled by Mean Shift filtering and the distinctions between the different clusters are performed satisfactorily.

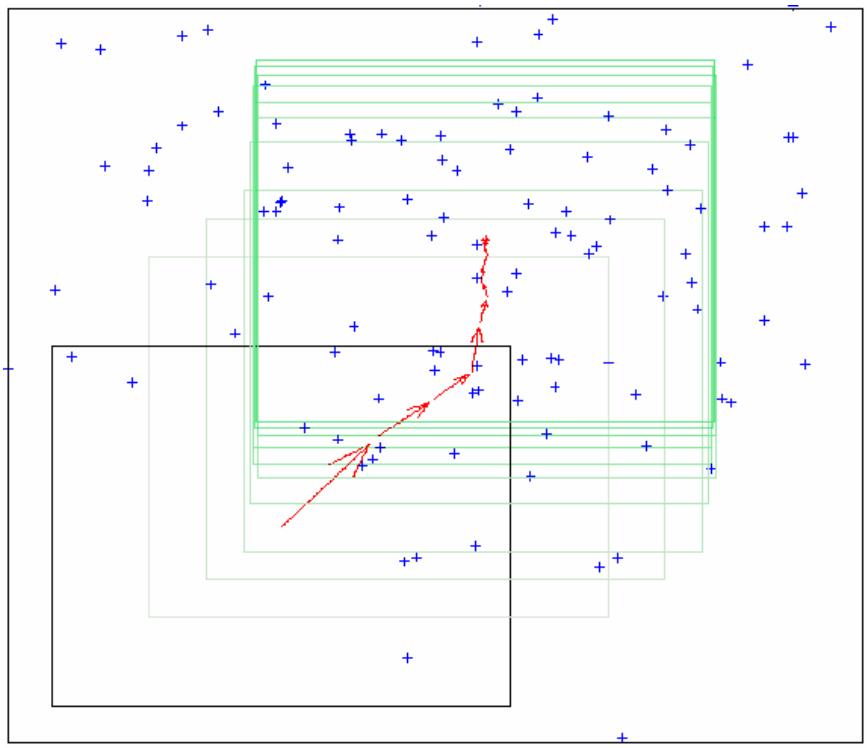


Figure 2.4: The iterative mean shift operation for a pixel [5].

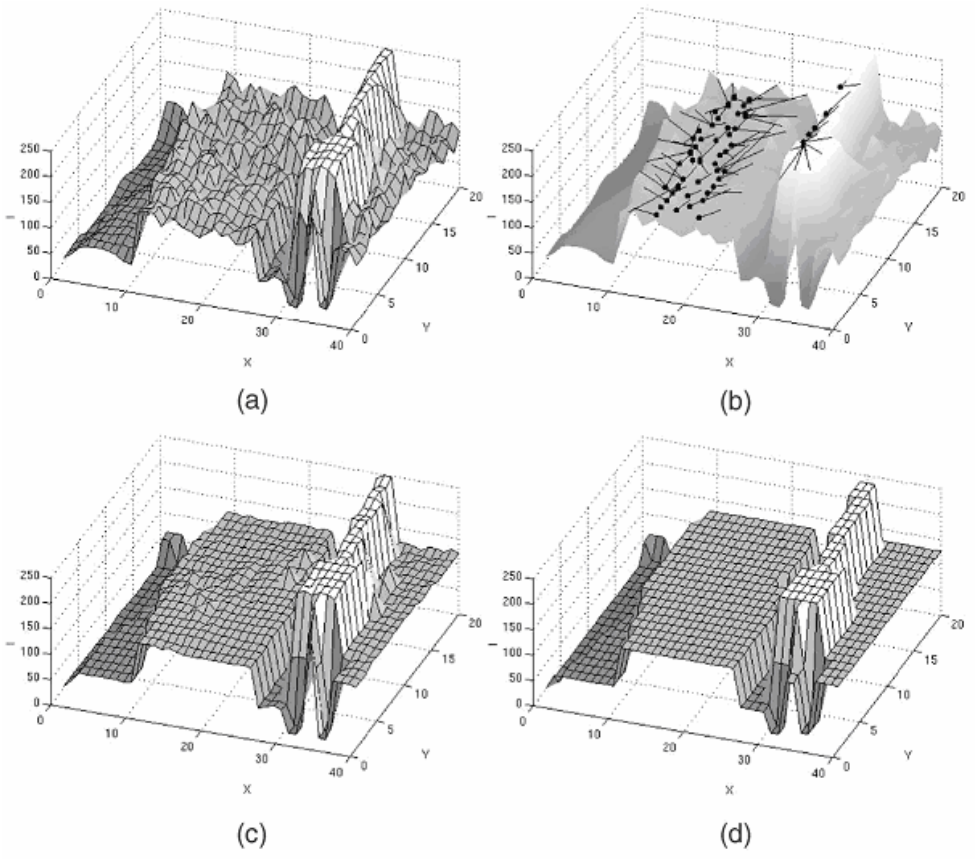


Figure 2.5: (a) The original data, (b) each pixel moves in the direction of the means and converges to the black points, (c) the smoothed data after mean-shift operations, (d) the segmentation result [4].

2.4 Graph Cut Image Segmentation

Normalized Cut Image Segmentation (NCIS) [6] is a global and a graph-based method that utilizes a splitting process beginning from the whole picture to the bottom as most of the segmentation methods based on graph cuts operate. The fundamental characteristics of graph-cuts based segmentation algorithms is the decomposition of eigenvectors of special matrices related with the constructed graph. A review of these methods that differ in terms of the matrices to be decomposed is given in [12]. Based on the comparison between three well-known segmentation methods based on eigenvectors, the performance differences depend on the statistical properties of the images such as color distribution [13]. Actually, NCIS method outperforms the other methods slightly for the overall case due to its normalization during the formulation of the segmentation problem.

There are two main steps in NCIS, the *construction of the graph* and the *iterative partitioning*. The top-down property of the normalized cut method is provided by initially mapping the image to a graph that holds the relations between pixels. An undirected weighted graph is constructed in which the vertices correspond to pixels and links correspond to weights evaluated via a linking cost function given in (2.3). In the cost function, I is the intensity image and $I(i)$ indicates the intensity value of i^{th} pixel and X indicates the location of the pixels. In the graph, the pixels which are located within a circle of radius R are linked to each other; hence the graph is partial and the link weights defines similarities between nodes as a function whose range space is [0,1].

$$w_{i,j} = \begin{cases} e^{-|I(i)-I(j)|^2/\sigma} & |X(i)-X(j)|^2 < R \\ 0 & elsewhere \end{cases} \tag{2.3}$$

The segmentation of an image is achieved by the division of the graph into smaller graphs which are disjoint by the cuts among the links iteratively. At each step, one graph (V) is partitioned into two sub-graphs, A and B, such that $V=A \cup B$, $A \cap B=\varnothing$. A cut on a link provides the separation of the two nodes connected to each other with the corresponding link. In NCIS algorithm, the cuts generally occur on more than one link and the separation of a group of nodes from another group of nodes is supplied, Figure 2.6. Every cut in the graph has a cost value evaluated by the summation of the link weights belonging to the removed (cut) links. In Figure 2.6**,** the cost of the cut is the summation of the link weights of 7, 9, and 12.

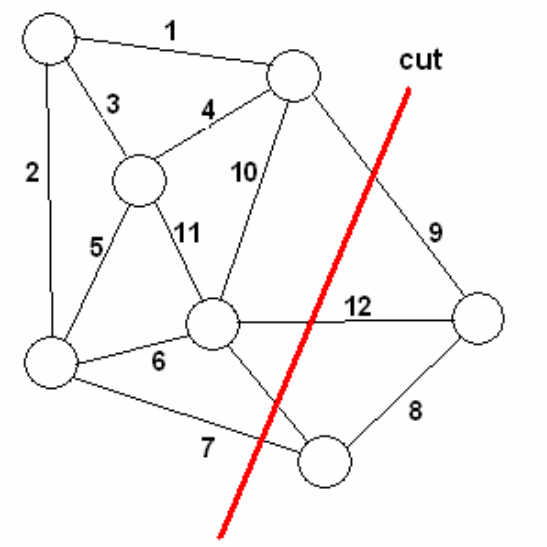


Figure 2.6: A sample cut on a graph

The eigenvector-based segmentation methods [6] [7] [8] try to minimize a cost function that is totally related with the cost of the cuts and the vertex weights during the partitioning process. The difference between those algorithms stems

from the difference in the cost function to be minimized; minimum cut method [7] aims to find the cut combination that will result in the minimum total cut cost as

$$Cut(A,B) = \sum_{i \in A, j \in B} w(i,j) \tag{2.4}$$

However, as explained in [6], such a minimization approach divides the graph into very small pieces, especially if there are vertices located at distant locations, resulting in an inferior partition. NCIS modifies the minimum cut approach by normalizing the cut costs with the total weights obtained by summing the total link weights of the nodes in the separated groups, as follows

$$Ncut(A,B) = \frac{Cut(A,B)}{TotalW(A,V)} + \frac{Cut(A,B)}{TotalW(B,V)} \tag{2.5}$$

where

$$TotalW(A,V) = \sum_{i \in A, j \in V} w(i,j) \tag{2.6}$$

In (2.5), *Cut(A,B)* indicates the cut cost evaluated between sub-graph *A* and sub-graph *B*. *TotalW(A,V)* indicates the total link weight between the vertices in *A* and the whole graph *V*. The normalized measure reflects how tightly the nodes within the disjoint groups are connected to each other.

The minimization of the normalized cut exactly is an *np-complete* problem; however, an approximate discrete solution can be obtained by formulating the problem into real value domain and using change of variables [6]. The theoretical analysis and the derivation of the new formulation [6] are explained in detail in the Appendix. This new formulation is given as

$$Ncut = \frac{y^T (D - W) y}{y^T D y} \tag{2.7}$$

where D is an NxN diagonal matrix of a graph with N nodes, indicating the total link weights belonging to each node individually. W is the affinity matrix of the graph showing the similarities between the nodes in a symmetric NxN matrix, in addition the diagonal entries of W are "***1***" since each node is totally similar to itself. Finally, y is a Nx1 matrix which is composed of real valued elements corresponding to the similarities of the nodes satisfying,

$$y^T D \bar{1} = 0 \text{ and } y(i) \in \{1, -b\} \tag{2.8}$$

The open form of b is given in Appendix, and $\bar{1}$ is an $Nx1$ matrix whose rows are all "1". The distinction of the values in y determines the partitioning such that the same signed nodes belong to the same group.

The expression in (2.7) is a Rayleigh quotient [14] and the minimization can be achieved by the solution of the generalized eigenvalue system [6]:

$$(D - W) y = \lambda D y \tag{2.9}$$

$$\Rightarrow D^{-1/2} (D - W) D^{-1/2} y = \lambda y \tag{2.10}$$

The second smallest eigenvector of the generalized eigenvalue system in (2.9) and (2.10) is the real valued solution of the normalized cut problem [6]. However, the solution is approximate, since y has different real valued entries after the eigenvalue decomposition that violates the second constraint in (2.8). The eigenvalue decomposition and determination of the second smallest eigenvector

provides a partitioning on the corresponding graph. In [6], this criterion is utilized recursively in order to perform the segmentation as follows:

1. Map the given image into a graph in order to relate the pixel (nodes) to each other via the similarity measures weighted on links.
2. Construct the D and W matrices through the link weights.
3. Solve the generalized eigenvalue system in (2.9) and (2.10) and determine the second smallest eigenvector.
4. Use the determined eigenvector to bipartition the graph.
5. Decide if the current partition should be subdivided and if necessary recursively partition the graphs.

In the fourth step of the proposed algorithm, the partitioning is performed by thresholding the eigenvector and labeling it into two portions, as the nodes above a threshold and the ones below the threshold. This threshold is determined by a one-dimensional search between the minimum and the maximum values within the entries of the eigenvector, and the partition that minimizes the normalized cut value given in (2.7). The decision of repartitioning the segmented regions depends on the minimum normalized cut value evaluated during the one dimensional search. If the value is small enough, then the repartitioning is not performed for the corresponding region. If it has a high value, then the recursive partitioning continues. A simple illustration of the partitioning process for a graph is given in Figure 2.7; the segmentation is performed iteratively grouping the nodes by minimizing the normalized cut value.

2.5 Modified Normalized Cuts Method

In NCIS algorithm, the constructed graph for a typical (200 x 300) image has 60,000 nodes and the affinity matrix is of size (60,000 x 60,000). Although, the

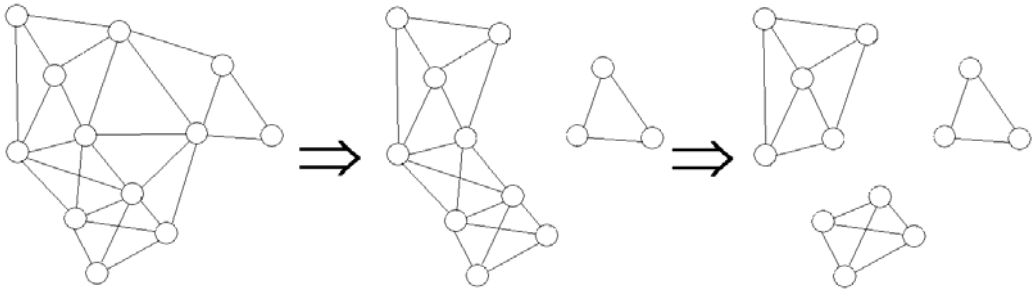


Figure 2.7: A sample recursive partition of a graph in three iterations.

affinity matrix is sparse and most of the entries are zero, the eigenvalue decomposition of such a matrix is time consuming. As the image size increases, the speed of the algorithm will be slower and the memory requirements may cause some problems. NCIS approaches to the problem of large sizes by down sampling the images and decrease the size of the affinity matrix considerably. Unfortunately, the down sampling operation causes the local intensity information to be lost and the object boundaries to be distorted at the final segmentation.

In this thesis, a modified normalized cuts algorithm is proposed in order to overcome the oversize problem without losing the local intensity information observed in down sampling. The graph is constructed from the regions obtained by the over-segmentation of the original image through Mean Shift [4], so that the local properties are embedded to the local regions and the number of the vertices in the graph is decreased to hundreds. The new graph structure (for the red squared region) is illustrated with an example in Figure 2.8. The random colored image corresponds to the over-segmented regions in the square and the centroids of these regions are utilized as the nodes of the graph.

Each node in the graph represents a small group of pixels, S_i, and the mean intensity of S_i is utilized as the characteristic of the node. The link weights are calculated via the linking cost function (2.3) described in NCIS method. The new representation of the graph structure causes an irregular distribution of the nodes among the graph as shown in Figure 2.9. The reason of such a node distribution is due to the different sizes of S_i's, as observed in Figure 2.8.

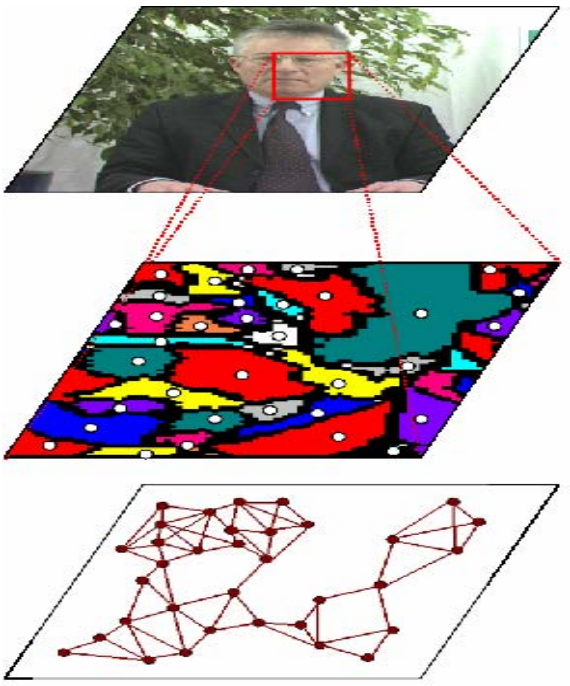


Figure 2.8: The modified graph structure

The irregular distribution of S_i's on the graph introduces a tendency to group the regions having more connections together without checking the similarities in between. This tendency is based on the formulation of the NCIS; when the minimization problem is analyzed it is observed that the normalization parameter

TotalW(A,V) is dependent on the total link cost which is obtained for each node by the summation of equal number of link weights for the regular graph.

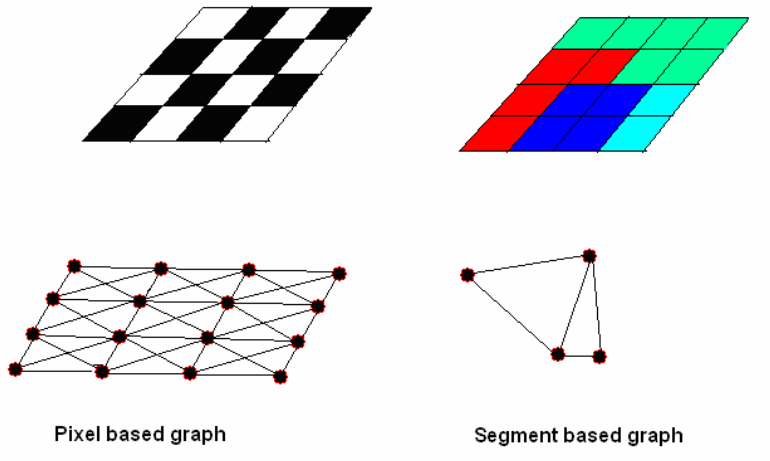


Figure 2.9: Segment based graph has an irregular structure.

Therefore, each link weight affects the partitioning process with the same ratio. In the irregular distribution however, since the number of links for each node differ, the nodes with more links have more terms in the summation and will result in a higher confidence, although the link weights are not strong enough. This phenomenon surpasses the weak links over the strong links, if they belong to a node having higher number of links and causes instabilities in the decomposition of the eigenvectors which results in an erroneous segmentation, although the NC is minimized.

The irregular distribution can be handled by enforcing each node to have same number of links. However, since the graph is undirected and the affinity matrix is symmetric, it is not easy to implement such an approach. The equalization method can be approximated by limiting the number of links for each node and allowing less variation on the distribution of the number of links. The limitation provides nodes to have similar number of links which can remove the suppression on

strong links belonging to nodes with less links. Although the solution does not force nodes to have same number of links, it dramatically decreases the irregularity.

2.6 Experimental Results

In this section, comparison of the algorithms is performed on different images and the effects of the modification on the NC segmentation algorithm are analyzed in detail. The first image utilized during the experiments is given in Figure 2.10 with its ground truth for segmentation. *Mosaic* [3] image is complicated in the local sense; however, the global characteristics help to separate the regions having different textures. The segmentation results of RSST, MS, NC and the modified NC are given in Figure 2.11. As it can be observed, the modified NC outperforms other methods and segments the image similar to the ground truth for segmentation. RSST also gives an acceptable segmentation output compared with the other local method MS. When the segmentation result for MS is analyzed, the tendency to smooth the similar textured regions can be observed clearly. The second image (*Cow* [3]) in Figure 2.12 has different characteristics compared to *Mosaic*; the local methods perform better segmentations than the global methods. However, the difference is not significant enough; in addition the proposed modified NC method refines the normalized cut algorithm and segments the image similar as the local methods. The *Objects* image in Figure 2.14 is best segmented with the proposed method, since the details of the cup object in the scene can also be observed although it is not differentiated in the ground truth, as shown in Figure 2.15. RSST also performs quite well compared with the other local segmentation method, MS. The refinement of the segmentation is clearly observed among the global techniques, as the proposed method preserves the local characteristics during the global minimization. For the *Baseball* image in Figure

2.16 and Figure 2.17, the performance of MS and RSST is better compared to the graph-based methods.

Up to this point, the visual quality of the segmentation methods have been measured in subjective terms; however, for an objective comparison the Mean-Square error values of the resultant segmentations should be utilized with respect to the ground truth. MSE is evaluated as follows:

$$MSE = \frac{1}{mxn} \sum_{i=0}^{mxn} \left| G(i) - G^k(i) \right|^2 \tag{2.11}$$

where G is the ground truth segmentation mask of the image and G^k is the segmentation mask of the k^{th} method. The resultant MSE plots for the images above are given in Figure 2.18.

According to the MSE plots, the proposed segmentation method has the minimum MSE values for the *Mosaic* and the *Baseball* images, the Mean-Shift segmentation is best for the image *Objects* and NCIS has the minimum MSE for the *Cow* image. In the overall case, the proposed method has the minimum total MSE for the images given above. Hence, the proposed algorithm provides some refinements over the Normalized-Cut segmentation method in both visual and MSE criteria. In addition, considering the well-known local methods of RSST and MS, the proposed algorithm is compatible with the state of art segmentation algorithms and unites the advantages of the local and global methods.

In eigen based approaches, the eigenvectors indicate the regions to be segmented according to different values assigned to its entries. A sample plot of the second eigenvector for the *Objects* image is given in Figure 2.19, the eigenvector distribution among the pixels indicate the regions to be separated by the different value levels. In the eigen-plots, the red colored pixels have higher values in the eigenvector and the blue pixels have smaller values.

In the proposed method, due to the irregular distribution of the nodes corresponding to small patches, limiting the number of links for each node is performed. Such a limitation provides the tendency of grouping the nodes having higher number of links to be removed to some extent and a better segmentation of the image. The second and third smallest eigenvectors of the *Mosaic* image is given in Figure 2.20, for the unlimited link and the limited link cases of the modified NC. In this figure, (a), (b) correspond to the 2^{nd} and 3^{rd} smallest eigenvectors for the case where the link number is limited. The regions to be segmented are clearly observed with high value changes at the actual object boundaries; however for the unlimited case, the eigenvectors in (c) and (d) do not have the segmentation information properly. The reason behind this fact is that, the nodes of the graph are distributed irregularly; hence the regions to be separated are determined according to the number of the links instead of the strength of the links. The segmentation according to the extracted eigenvectors is given in Figure 2.21 for both of the cases. The increase in the performance of the segmentation is clearly observed, the segmentation for the limited case is more realistic and closer to the ground truth than the partitioning without the limiting the number of the links for each node. Thus, the modification in the construction of the affinity matrix improves the segmentation results.

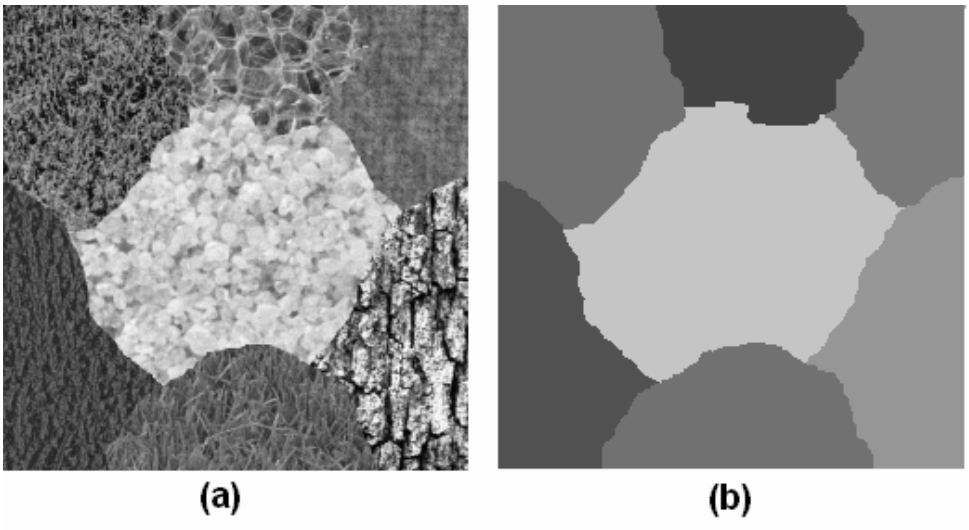


Figure 2.10: (a) the *Mosaic* image, (b) the ground truth segmentation.

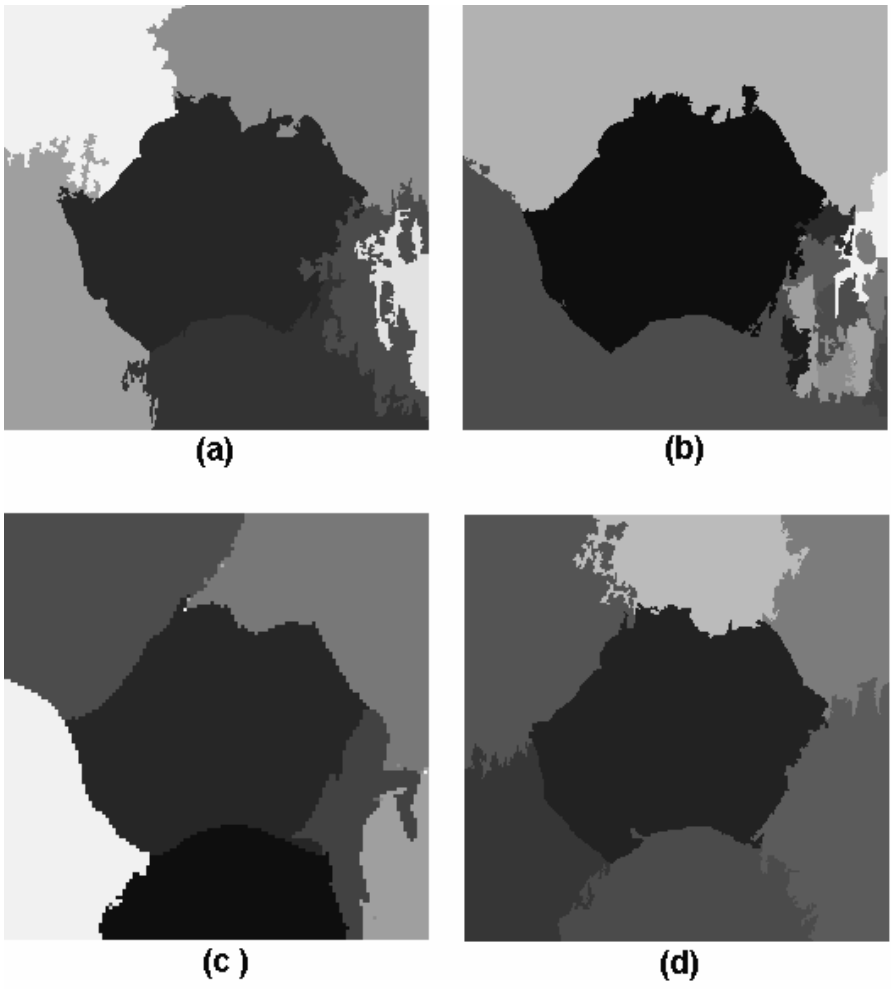


Figure 2.11: The segmentation results of *Mosaic* (a) RSST, (b) MS, (c) Normalized Cut, (d) Modified NC.

Figure 2.12: *Cow* image (a) the color, (b) ground truth segmentation.

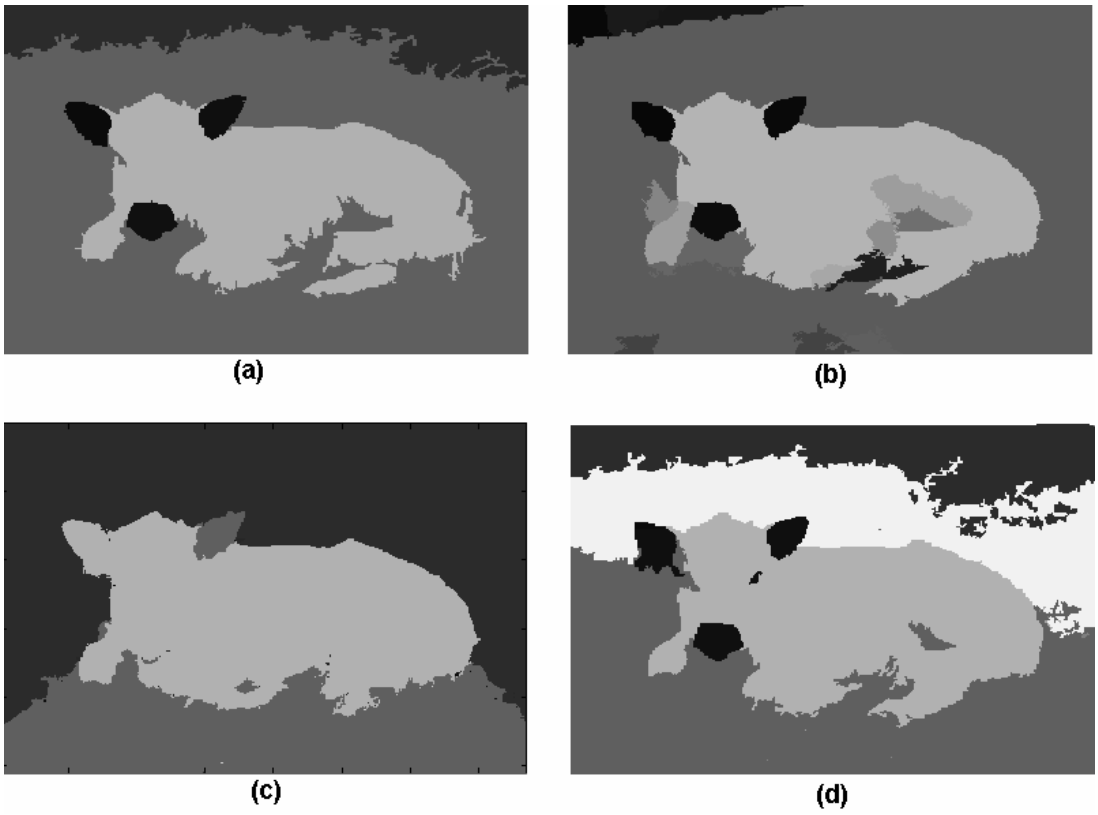


Figure 2.13: The segmentation results of *Mosaic* (a) RSST, (b) MS, (c) Normalized Cut, (d) Modified NC.

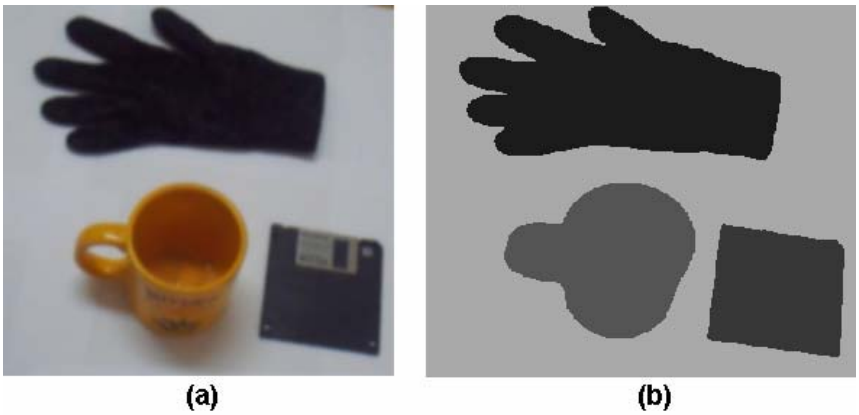


Figure 2.14: (a) The *objects* color image, (b) the ground truth segmentation

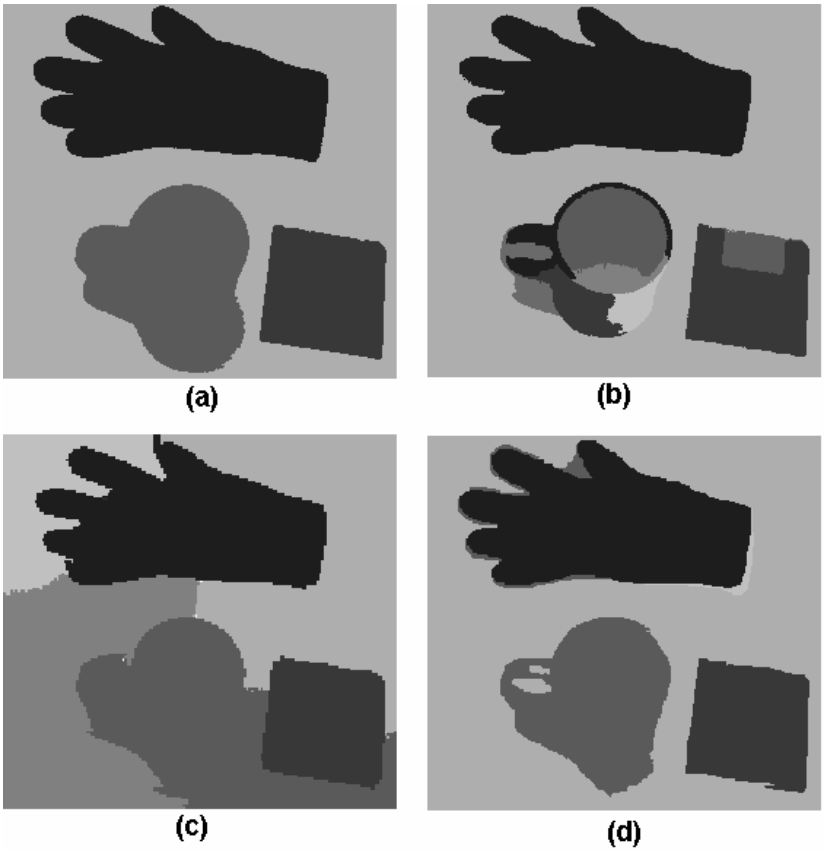


Figure 2.15: The segmentation results of the *object* image (a) RSST, (b) MS, (c) Normalized Cut, (d) Modified NC.

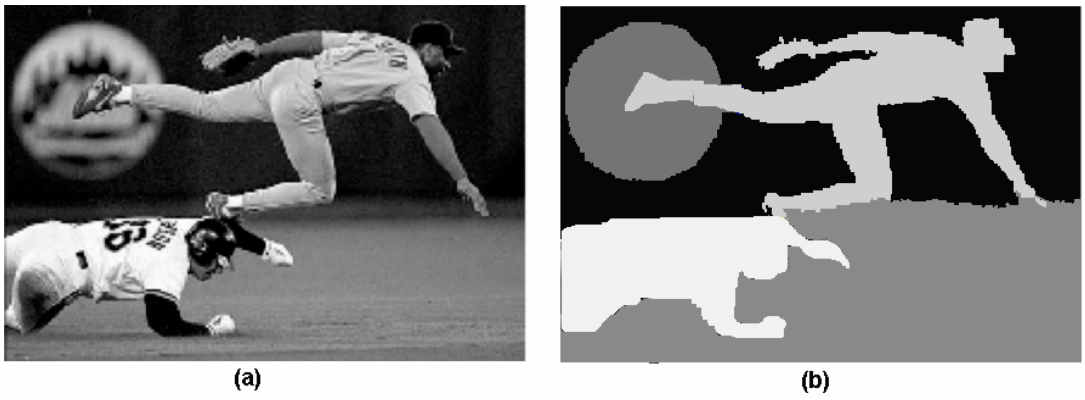


Figure 2.16: The *Baseball* image with its ground truth segmentation.

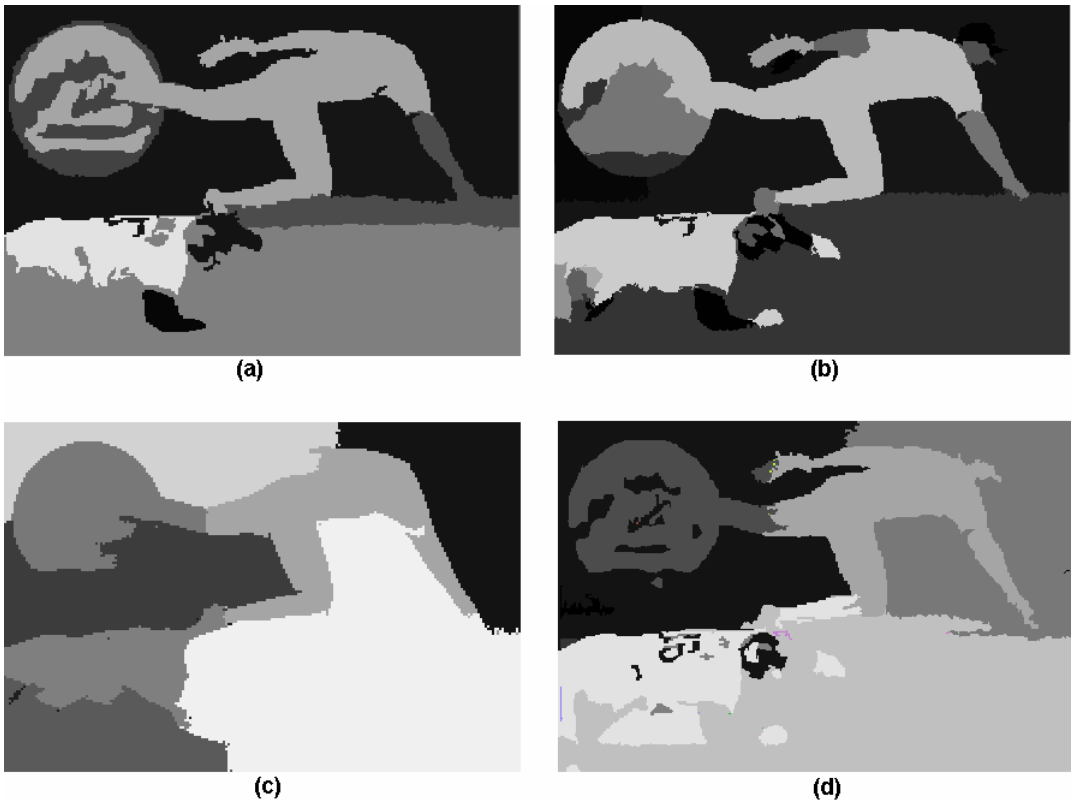


Figure 2.17: The segmentation masks of the *Baseball* image (a) RSST, (b) MS, (c) Normalized Cut, (d) Modified NC.

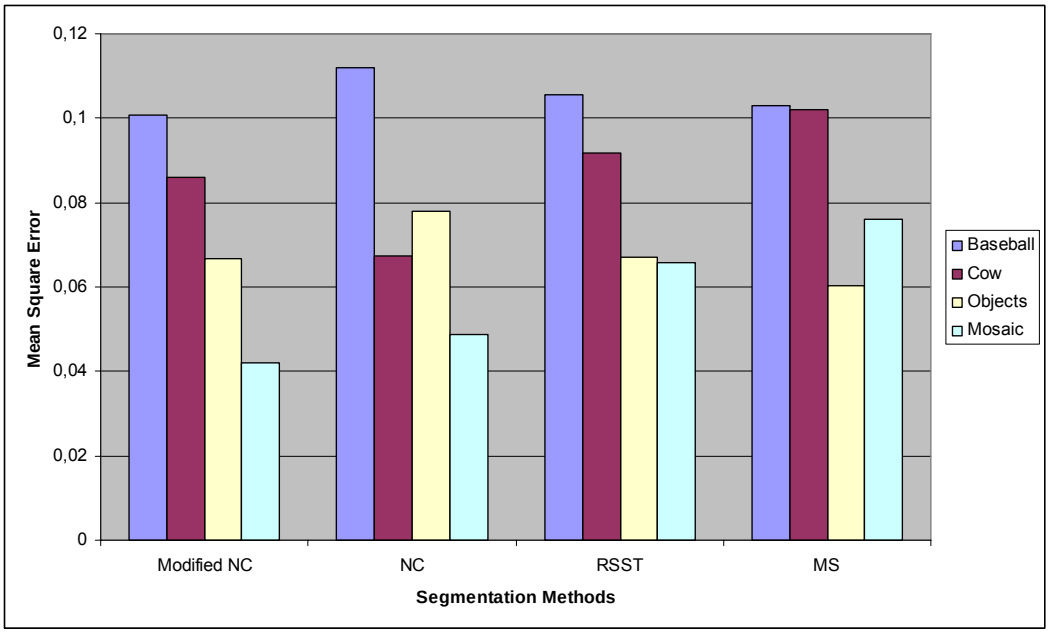


Figure 2.18: The Mean-Square error plots of the algorithms for the images above

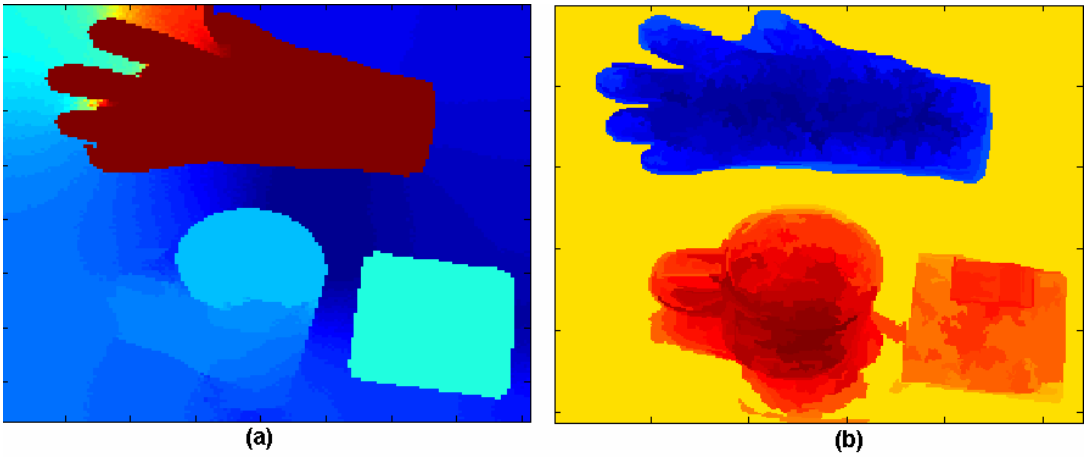


Figure 2.19: (a) The plot of the second smallest eigenvector with NC method, (b) with modified NC method. The red colors indicate higher values and blue colors indicate smaller values.

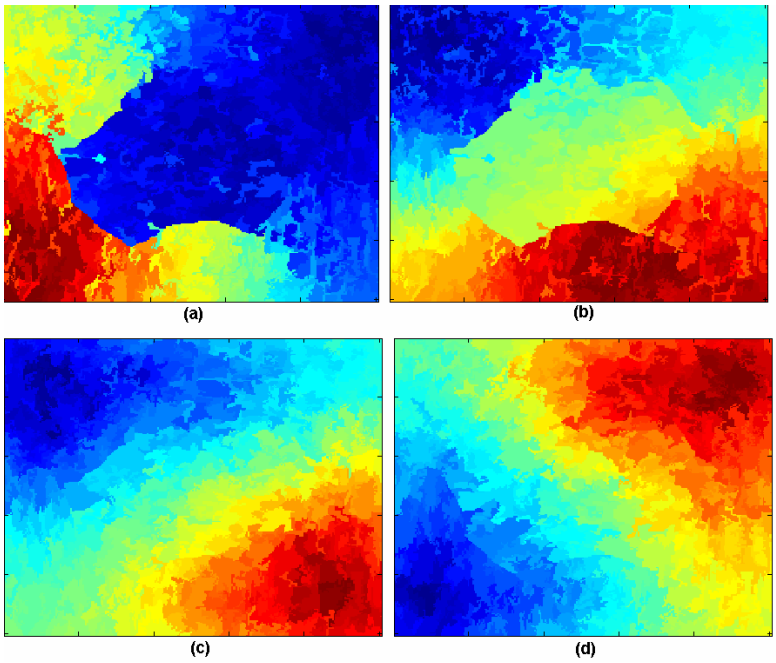


Figure 2.20: (a), (b) the second and the third smallest eigenvector of the *Mosaic* image with the number of links limited; (c), (d) the eigenvectors for the unlimited case.

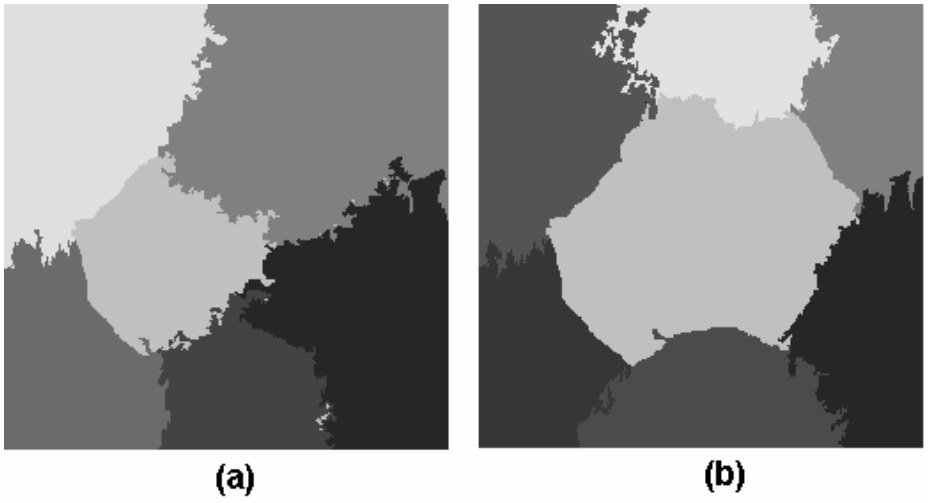


Figure 2.21: The segmentation results of the *Mosaics* (a) unlimited link, (b) limited link

CHAPTER 3

DENSE DEPTH FIELD ESTIMATION

This chapter starts with a brief introduction about the geometric relations between 3-D scene points and their projected 2-D images. This background material is followed by a literature survey on dense depth estimation methods. In the following sections, a novel algorithm is presented in three main parts. In the first part, segment-based stereo matching is explained which involves plane- and angle-sweeping operations. In the following part, this algorithm is extended to multi-view case by exploitation of existing tools in stereo matching. The refinement methods, which are based on segment split algorithm and pixel-based optimization via belief propagation, are discussed in the next section. Finally, experimental results are given on different type of data sets and the reliability of the algorithms is discussed.

3.1 Camera Model

The observation of a scene through an imaging device can be modeled via a transformation from 3D world coordinates to 2D image coordinates. Except for certain special cases, the corresponding transformation is non-linear due to the *radial lens distortion* [16] resulted from the physical structure of the lenses. However, the non-linear characteristics of the transformation can be removed by

pre-processing or be neglected and the basic linear pinhole camera model is obtained.

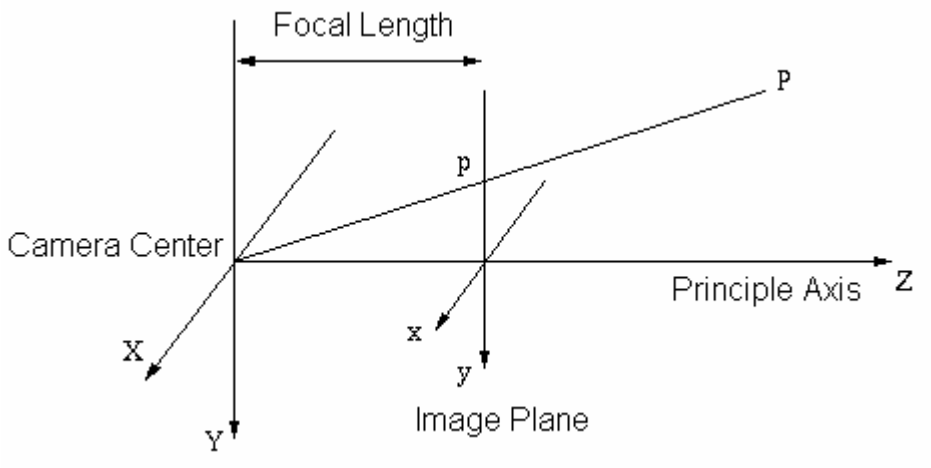


Figure 3.1: Basic Pinhole Camera Model

In basic pinhole camera model, a point in 3D world is projected onto 2D image plane by the intersection of a line passing through the 3-D point and a fixed location, called as the *camera center,* which is located at focal length (f) distance from the image plane. Such a transformation from 3D to 2D can be represented by a 3x4 matrix which is denoted as *camera projection matrix* [16].

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.1}$$

In equation 3.1, the 3D world and the camera coordinate systems are located on the same Euclidean system, however in general case; there is also a

transformation between these two coordinate systems. Therefore, the *camera projection matrix* should be generalized by applying a rotation and a translation transformation between two coordinate systems. The inclusion of this transformation leads to the following form

$$x = PX = K[R \mid t]X \tag{3.2}$$

where K matrix (3x3) is the intrinsic parameters derived via focal length and internal calibration information; R matrix (3x3) is the rotational and t (3x1) matrix is the translational transformation among Euclidean systems, known as exterior parameters. X denotes the point in 3D world coordinates and x is the pixel coordinates on the image plane. A camera is denoted as *calibrated*, if both exterior and the interior parameters are known a priori.

3.2 Epipolar Geometry

Two arbitrarily located cameras in a 3D scene define a geometric relation, known as *epipolar geometry*, which has specific properties. In Figure 3.2, the elements of the epipolar geometry, such as *epipolar plane*, *baseline*, *epipole* and *epipolar lines*, are illustrated. The plane defined by a 3D point and two camera centers is denoted as *epipolar plane*, whereas the *baseline* is the line connecting two camera centers and the *epipoles* are the intersections of the baseline with the image planes. *Epipolar lines* are the intersections of the image planes with the epipolar plane and all of the epipolar lines pass through epipoles [16].

Each point in one image is mapped to a distinct epipolar line in the other image; hence they are quite important in computer vision. This property is illustrated in Figure 3.3 with an example indicating the role of epipolar lines in 3D information extraction. In Figure 3.3, the back projection of the point x in *camera*-1 to 3D and

its corresponding projections to the image plane of *camera*-2 are shown. By using the epipolar relation between two images, the 3D position of a point in one of the images can be extracted, if its correspondence is known at the other image.

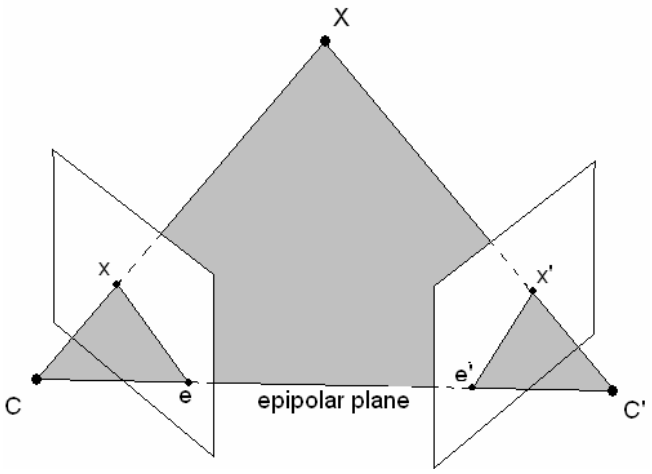


Figure 3.2: The epipolar geometry

In addition to epipolar geometry, there is another special transformation between a camera and a plane in 3D. This transformation can be denoted as *homography* which is defined as the point-to-point mapping between a 3D point on a ground plane and its observed 2D point in image plane, Figure 3.4. The homography matrix can be derived from the given projection matrix (3.3) and the plane equation of a 3D plane (3.4).

$$x' = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} A & B & C & D \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \tag{3.4}$$

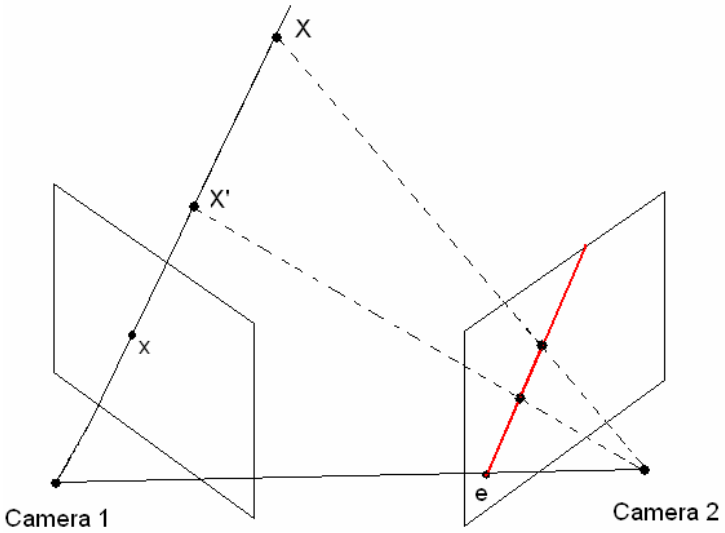


Figure 3.3: The correspondence of x is on its corresponding epipolar line which is shown in red

By writing the value of Z in terms of plane parameters and other independent variables, as

$$Z = \frac{-(AX + BY + D)}{C} \quad \text{э } C \neq 0 \tag{3.5}$$

Then the image formation equation can be rewritten as follows

$$x' = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ \dfrac{-(AX+BY+D)}{C} \\ 1 \end{bmatrix} \tag{3.6}$$

The third variable in the last term of (3.6) can be removed with the transformation of the projection matrix into a 3x3 matrix by embedding the third column into the other columns according to plane parameters, as

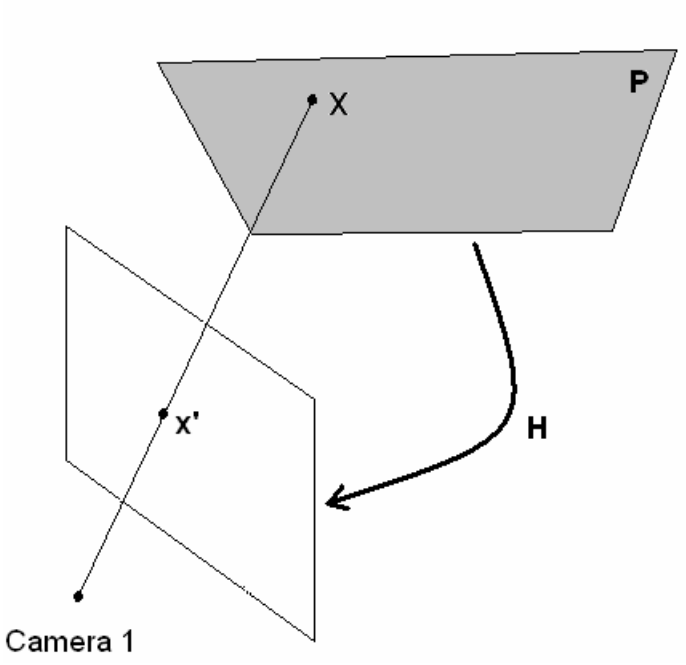


Figure 3.4: The planes in 3D define homographic relation between different cameras

$$x' = \begin{bmatrix} (P_{11} - \frac{AP_{13}}{C}) & (P_{12} - \frac{BP_{13}}{C}) & (P_{14} - \frac{DP_{13}}{C}) \\ (P_{21} - \frac{AP_{23}}{C}) & (P_{22} - \frac{BP_{23}}{C}) & (P_{24} - \frac{DP_{23}}{C}) \\ (P_{31} - \frac{AP_{33}}{C}) & (P_{32} - \frac{BP_{33}}{C}) & (P_{34} - \frac{DP_{33}}{C}) \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ \frac{-(AX + BY + D)}{C} \\ 1 \end{bmatrix} \quad (3.7)$$

Hence, the reduced matrix forms the homography matrix between the camera and the 3D plane, (3.8).

$$\therefore x' = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.8)$$

3.3 Literature Review for Dense Depth Estimation

Dense depth-map estimation has attracted many researches due to its wide application areas in computer vision, such as 3D object modeling [17], segmentation and image-based rendering [18]. There are mainly two approaches for dense depth estimation, *stereo* [20] and *multi-view* [51] matching. In stereo approach, the images taken from two horizontally aligned cameras are utilized to estimate the horizontal shifts, or *disparities* of each pixel. This system is similar to the human-vision system, with horizontal placement of the cameras as the eyes. Two views can also be aligned horizontally by rectification [19], if the cameras are not parallel to each other. An excellent taxonomy of the stereo algorithms is given by Szeliski et.al in [20]. On the other hand, in multi-view approach, the matching is performed among multiple images [21]-[25]. Although, the past research efforts focus on stereo matching more, the advances in capture technology and the emerging new applications, such as free-view TV or 3DTV,

that require multiple images, has resulted in a tendency towards the multi-view matching.

Most of the dense depth estimation algorithms approach the problem by making use of two basic assumptions, namely the *smoothness* of the depth field and the *high level of visual similarity* between image neighborhoods of corresponding pixels. The solution for the dense matching problem is usually classified into two groups, as *local* and *global* techniques [20], in both stereo and multi-view cases. Both classes utilize the high-level visual similarity by enforcing the matches to have similar intensity values or variation within a kernel.

In local methods [26]-[29], the smoothness of the depth map is imposed explicitly and a "winner-take-all" style optimization utilized generally. The matching cost function is aggregated by summation or averaging over a support region. The window size determines the smoothness: As window size increases the depth map becomes smoother. In [26], the window size is modulated with respect to the intensity gradient in order to use the observed data more efficiently. The advantage of the local methods is their real time performance; however they do not yield high quality depth maps as global methods do, since they are strongly dependent on the intensity values only and the uniqueness of the matching is not enforced for all of the images.

In global methods [30]-[37], the smoothness constraint is utilized implicitly by enforcing neighboring pixels to have similar depth values. These methods are formulated in an energy-minimization framework and the objective is to optimize the global energy for the estimated depth map. Belief propagation via Markov Random Field formulation [30][31][32], graph cuts [33][34][35] and dynamic programming [36][37] are the most common optimization methods used in dense depth estimation algorithms.

Dense depth estimation is a challenging problem due to scene complexity and the limitation of the observed data. The complexity problem is tried to be solved by some assumptions about the scene, such as planarity [38], slanted surfaces [39], and ordering constraint [36], however, all of them indicate specific scene models and will not cover all kinds of scenes. Moreover, the occlusions limit the observation of the data, and this problem is usually approached by increasing the number of the images taken, as in the multiple-view scenario. Since dense depth estimation algorithms rely on high visual similarity, they especially fail at un-textured regions, where the color variation is less, occlusions and object boundaries where depth discontinuities are observed.

In the last decade, many region-based dense depth estimation algorithms [38], [40]-[49] resulting in high quality depth maps have been proposed in order to handle depth discontinuities and occlusions, as well as preserve object boundaries. The importance of these algorithms is due to the fact that they are neither local nor global, although they combine the advantages of both of these methods into a single approach efficiently. These algorithms rely on the assumption that the scene is composed of small non-overlapping planes, all of which correspond to distinct segments obtained via grouping pixels of homogenous color. Hence, smoothness constraint is valid within each segment and depth distribution is allowed to change sharply between segment boundaries, which generally correspond to object boundaries as well.

Region-based stereo matching algorithms mainly consists of 4 steps: In the first step, the reference image is over-segmented, in order to obtain non-overlapping plane masks. After the segmentation stage, the initial depth map is estimated by local stereo matching algorithms for both of the images and the reliable points are determined. In the next step, plane parameters are estimated for each segment by the help of the reliable points. In the final step, the depth (disparity) distribution between segments is determined by different optimization techniques, such as greedy search [38], [45], belief propagation [40], graph-cuts [42]. The initial

segmentation step provides the detection of regions where the planarity assumption is valid and distinguishing the regions located at object boundaries and having sharp intensity changes. Therefore, in order to estimate the depth discontinuities at object boundaries, the segmentation information becomes crucial.

Apart from the algorithms mentioned above, the algorithm of *Microsoft Research Group* [46] proposes a different approach. In [46], there is local matching in the segment domain, while treating them as point regions (super pixels) to be matched, instead of local matching in the pixel domain, as in the second step of region-based algorithms. In addition, [46] utilizes a different optimization technique and aims to find multiple depth fields corresponding to different images used during the depth estimation at the same time. Such an approach is similar to the proposed algorithm in the next section, in terms of segment matching and its extension to the multi-view.

Although, they result in realistic depth maps, segment-based stereo matching algorithms are generally studied in the context of narrow-baseline stereo and for almost fronto-parallel cameras, except for the method in [46] so far. Hence, they are also expected to suffer from wide-baselines and occlusions, as most other stereo algorithms. In addition, considering the local stereo matching step of these algorithms, it is obvious that they could also suffer for the untextured and large regions, since they utilize pixel matching in local domain which might be deceptive. Region-based algorithms perform better, where the scene is composed of almost planar surfaces; however for more natural scenes including round shaped objects, randomly changing surfaces; they might fail due to the planarity assumption.

3.4 Proposed Stereo Dense Depth Map Estimation Algorithm

In this work, a novel region-based approach to the dense depth estimation problem with some basic differences from the traditional region-based approaches is proposed. Special attention is devoted to overcome the difficulties resulting from large untextured regions, considerably wide-baseline views, and the effects of enlarging or shrinking of some regions due to rotation of cameras which validates the fronto-parallel positioning of stereo cameras. In the proposed method, each segment that is obtained via color *over-segmentation* of the images is utilized to extract the "primitives" to be matched instead of the individual pixels. The term over-segmentation implies the action of obtaining an image with relatively high number of regions (relative to the semantic objects in the scene and total number of pixels) as a result of any segmentation method. Therefore, large and untextured regions are expected to be handled relatively easily, since local matching of pixels, which is not reliable at the untextured regions, is avoided. The proposed approach provides a strict relation between the pixels belonging to the same segment.

The algorithm also provides a multi-view extension, which is important to obtain more reliable depth maps by increasing the number of the observed views. The major step of the proposed method is plane and angle sweeping of the segments in 3D during the matching process of the segments. In the angle sweeping stage especially, the enlarging and shrinking of the regions are also taken into consideration according to the camera positions. Thus, the slanted planar regions are handled as well.

The fundamental assumption for all segment-based dense depth estimation algorithms is the scene planarity which is often acceptable in man-made structures. However, this assumption is not valid in every scene that makes this

approach less desirable. In order to relax the planarity assumption and estimate the non-planar objects, such as balls, trees, cones; finally a pixel-based refinement could be performed, that results in reliable and more realistic depth maps.

3.4.1 Definition of Mathematical Expressions and Overview of the Algorithm

In order to clarify the algorithm, the utilized variables and functions should be explicitly defined. In this content, a color image is defined by I, whereas a depth image is denoted by DI. As a result of any type of segmentation on I, the i^{th} segment is defined as S_i, the number of pixels in S_i is given as N_i and the set of the boundary pixels is expressed with B_i. The dense matching is considered as a labeling function, $L(S_i)$, which assigns planes ($D_i(\alpha,\beta)$) to the segments, S_i. The union of these disjoint segments forms I, simply as

$$I = \bigcup_{i=1}^{N} S_i \quad \ni \quad S_i \cap S_j = \varnothing, \ \forall i \neq j \ and \ S_i, S_j \in S \tag{3.9}$$

Moreover, an arbitrary region in an image is denoted as R_i. D_i defines the i^{th} depth plane and $D_i(\alpha,\beta)$ defines the rotated i^{th} depth plane by the angles α,β around x- and y-axes, respectively. Finally, the set of neighboring segments for S_i is defined by NB_i.

Considering the general structure of region-based dense depth field estimation algorithms, which result in accurate depth fields, there are mainly four steps and the distinctions between these algorithms are mostly generated from different optimization techniques at the final stage. In the proposed approach [49], the number of steps is decreased to three by merging the second and third steps of the traditional algorithms with some distinctive methods, such as plane- and angle-

sweeping. In order to perform these sweeping methods, the epipolar geometry between the cameras should be known. Hence, the proposed stereo algorithm takes two images and the associated epipolar geometry as input and determines the depth field of the image which is assumed to be the reference view.

In the first step of the algorithm, both of the stereo images are over-segmented in order to determine regions which correspond to planar patches. In the following step, the initial depth map of the reference view is estimated via plane, angle sweeping and left-right consistency check between both of the views. Finally, an iterative update of the segment location and rotations in 3D is provided via a greedy search algorithm, making use of visibility and reconstruction quality constraints which finalizes the estimated depth map. The flowchart of the algorithm is given in Figure 3.5 and an example is illustrated with the steps of the algorithm in Figure 3.6. In the following sections, each of these steps is explained in detail.

3.4.2 Over-Segmentation of Input Images

The main assumption of the algorithm is modeling of the scene via planar 3-D patches. For the extraction of these patches and a reliable planar model, these planar regions are assumed to correspond to the segments, S_i, of homogenous color in the images, which usually belong to the same object. Such an approach provides similar depth values for the similar colored neighboring pixels.

In order to model the scene properly, the extracted segments should involve similarly colored pixels and preserve the local variations in a region, thus the over-segmentation becomes a crucial step. If the number of extracted segments,

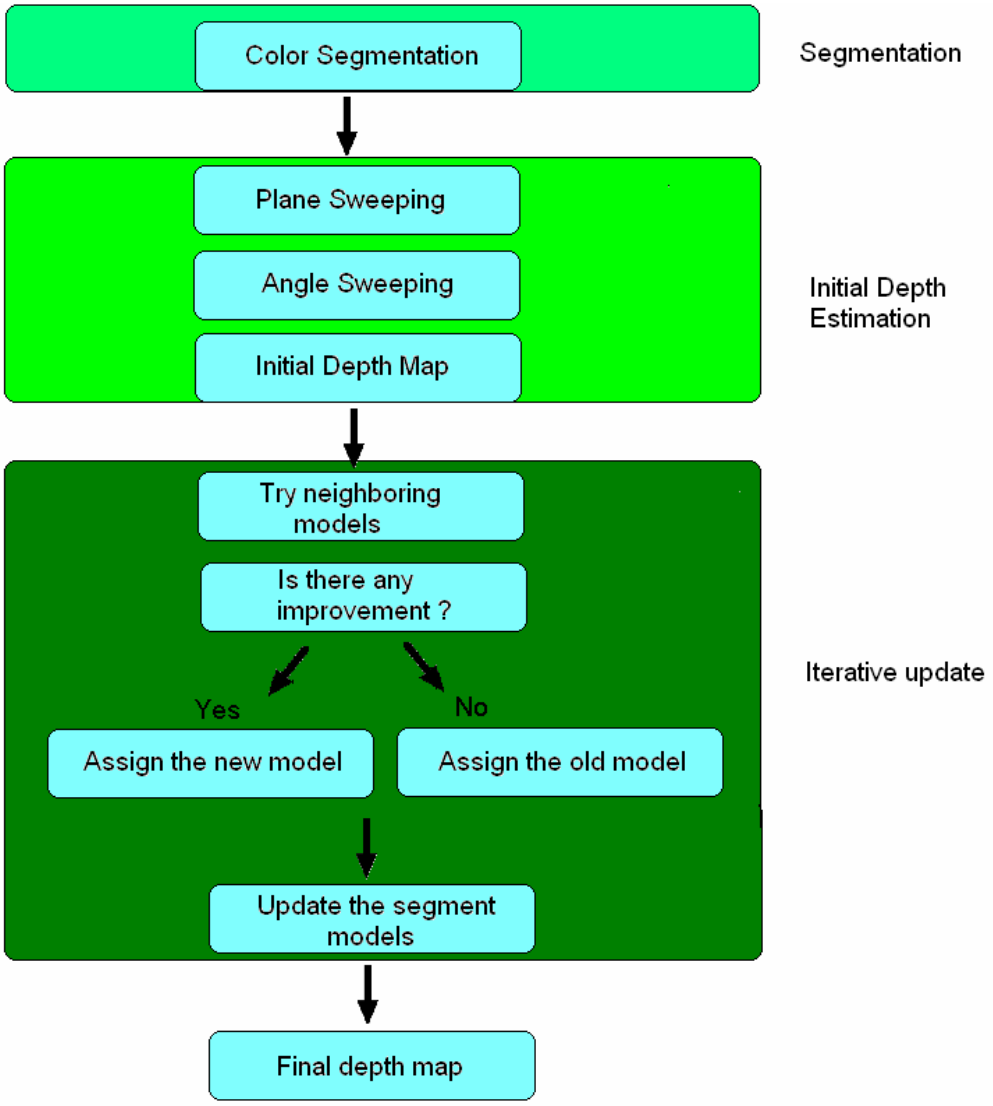


Figure 3.5: The flowchart of the proposed algorithm.

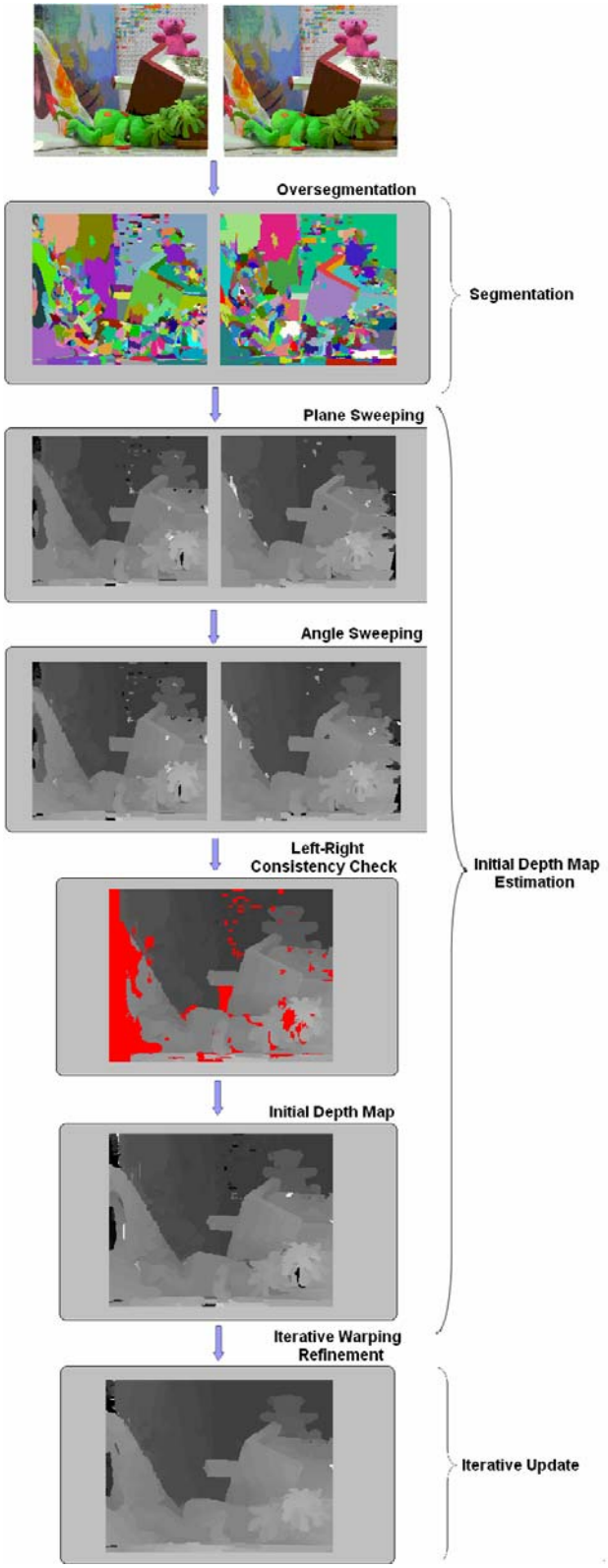


Figure 3.6: The steps of the algorithm on a sample stereo image.

N, is low, then the local intensity variation might be lost. On the other hand, if it is high, then the algorithm converges to pixel-based methods and the local information is utilized inefficiently. In this work, two different color segmentation algorithms with different characteristics have been employed and the effects of different segmentation results are examined. As mentioned in the prior chapter, the utilized segmentation algorithms are Recursive Shortest Spanning Tree (RSST) [3] and Mean Shift (MS) [4]. The randomly colored over-segmented images via RSST and MS are presented in the experimental results section. From these figures, it can be easily observed that MS is insensitive to small intensity gradient changes, while RSST divides similar colored regions, even if there is a slight gradient change in intensity. Hence, in the following steps, MS is utilized as the initial over-segmentation algorithm.

The planarity assumption of the scene enforces depth variation within each S_i to be smooth, since each segment defines a plane in 3D. Besides, the smoothness is also valid between neighboring segments, NB_i, which have similar color distributions. Hence, depth discontinuities are expected to be observed only at the segment boundaries, which have high intensity differences. This approach preserves depth discontinuities at the object boundaries, as long as there is an intensity gradient among the pixels in NB_i, which is the usual case. Finally, the over-segmented regions are taken as input for the following initial depth map estimation step.

3.4.3 Initial Depth Map Estimation via Plane and Angle Sweeping

Upon the identification of the candidate planar regions via over-segmentation, the initial 3D model is constructed. When the related work on segment-based stereo is investigated in the literature, it can be observed that most of the algorithms perform a simple block matching in the pixel domain and then a plane-fitting

operation in order to estimate 3D plane models for each segment [40]-[49]. During the model estimation step, only the pixels, which are reliable in left-right consistency check, are utilized.

In the proposed method, a novel approach based on plane- and angle-sweeping is utilized instead of these two steps of the traditional methods. One of the novel parts of the proposed algorithm is performing a matching for a group of pixels (segment) to a region (R_i) as $S_i \rightarrow R_i$, instead of matching pixels individually. The new approach increases the robustness of the algorithm against noise and repeating regions, since it utilizes similarity of a group of pixels, instead of a sole pixel. Application of plane- and angle-sweeping for the segments individually is another novel approach of the algorithm which provides a plane search without violating the planarity assumption.

There are three main parts in the initial depth map estimation step. In the first part, plane sweeping is performed and D_i's are assigned for all S_i. In angle sweeping, the planes are rotated around x- and y-axis and the best orientation with the depth position are determined, $D_i(\alpha,\beta)$. The plane and angle sweeping are performed for both of the images and the 3D plane parameters are estimated for each segment in these images. At the final step, the reliable segments of the reference view are determined by a left-right consistency cross-check by the extracted plane parameters. For the unreliable segments, a final search is performed and the initial depth map estimate is obtained.

3.4.3.1 Plane Sweeping

The plane sweeping approach has been introduced by [51], in order to match object boundaries between multiple images. The main idea stems from the requirement of relating multiple edge images to each other and extracting depth

information. In this manner, a surface in the reference view can be easily back-projected to 3D space and by using the projection matrices, the corresponding locations of the surface in the other views are determined easily. The relation between images is defined via homographies which also determines the projective transformations between planes. In order to extract the homographic relations, the depth planes which are parallel to the reference image plane are defined at different depth values.

In this work, plane sweeping is utilized in order to assign D_i values for the segments, S_i. As indicated, the depth (i.e. distance from the reference image plane) is assumed to be constant among a plane and the scene is sampled with a number of planes that are parallel to the reference image plane, as in Figure 3.7. The distance between parallel planes are determined by the disparity range and the epipolar geometry between the reference view and the closest neighboring view.

Once the plane equations are determined, the relation between two views can be defined via homographies with the given projection matrices, which are explicitly given in the following equations:

$$H_i(d) = \begin{bmatrix} P_1^i & P_2^i & P_4^i \end{bmatrix} - P_3^i \begin{bmatrix} \frac{n_1}{n_3} & \frac{n_2}{n_3} & \frac{n_4}{n_3} \end{bmatrix} \tag{3.10}$$

$$H_{i,j}(d) = H_i^{-1}(d) H_j(d) \tag{3.11}$$

where P_j^i corresponds to the j^{th} column of the projection matrix of the i^{th} camera (I_i) and n_i indicates the plane normal parameters. In the equations, $H_i(d)$ defines the homography from the d^{th} 3D plane to the i^{th} cameras image plane and $H_{i,j}$ defines the homography from the i^{th} camera to the j^{th} camera. In the stereo case i, j are equal to 1 and 2, respectively (see Figure 3.8).

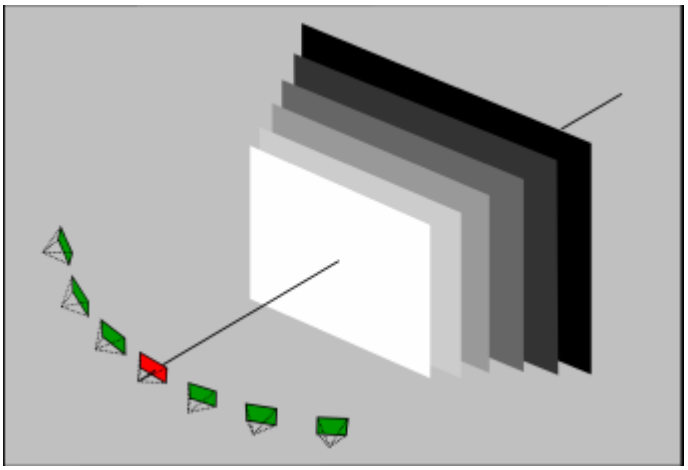


Figure 3.7: The space is divided into parallel depth planes perpendicular to the principal axis of the reference camera.

Each homography defines a one-to-one mapping function between two images, which is defined as follows:

$$f_H : I_{ref} \rightarrow I_2 \ , \exists \ f_H(S_i) = R_i \subset I_2 \tag{3.12}$$

$$Hx = x', \ x \in S_i, \ x' \in R_i \tag{3.13}$$

In the plane sweeping part, the aim is to estimate the best depth plane for each S_i. Actually, utilization of 3D planes is crucial, since the multi-view extension of the algorithm can be easily obtained. In addition, such a parameterization also yields rotation around x- and y-axis that is simpler to perform. After the homographies are extracted for different depth planes, *H(d)*, the depth values are assigned to the segments by minimizing a cost function based on intensity similarity measure between two views. For an arbitrary segment S_i, the cost function corresponding to the d^{th} plane can be written as:

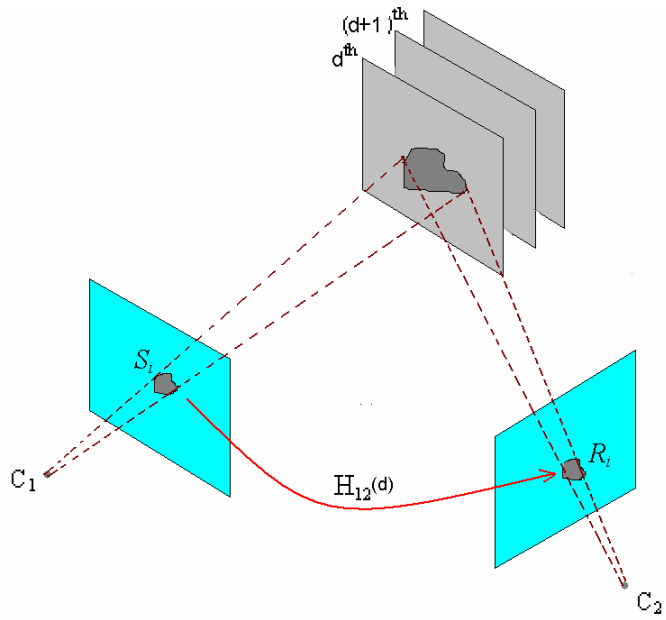


Figure 3.8: The homographic relation between two views via depth planes

$$C_{S_i}(d) = \frac{1}{N_i} \sum_{\substack{x \in S_i \\ x' \in R_{id}}} \left| I_{ref}(x) - I_2(x') \right| \tag{3.14}$$

When the cost function in (3.14) is analyzed, it can be observed that all the pixels have contribution to the cost value of the segment. However, when typical scene geometry is considered, there might be partially occluded sub-regions within some segments which will result in unreliable cost values. As illustrated in Figure 3.9, segment-*A* has an occluded region (black) which can not be observed by the other camera. Hence, the contribution from such an invisible part should be eliminated to form a more reliable cost function. The elimination is not trivial, since the visibility of the pixels can not be determined, unless all the neighboring depth values are known.

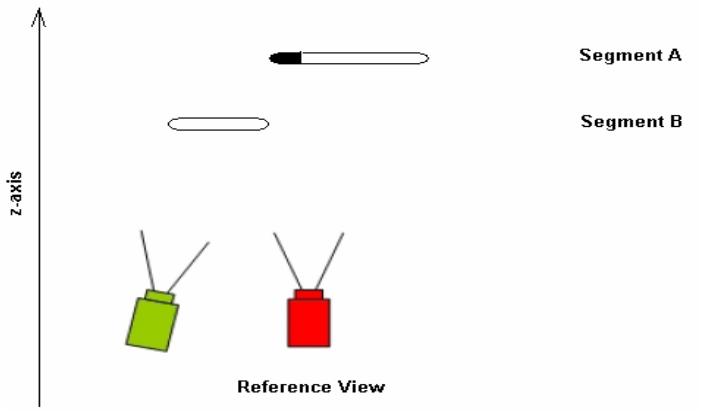


Figure 3.9: Typical example for partially occluded segments.

A simple idea is proposed in order to initially handle such cases at the plane- and angle-sweeping step. It is assumed that if the pixel similarity cost is above a threshold, than it should be in the partially occluded region within the segment Thus, the contribution of the pixel should be discarded in the summation term. Although the assumption is not valid for all cases, it provides some refinement and reliability in the cost function. Some tests have been performed in the experimental results section in order to compare the original cost function and the modified version, and the improvement can be observed clearly in the simulations. This new cost function could be written in the following way:

$$C_{S_i}(d) = \frac{1}{N_{threshold}} \sum_{x \in S_i} g_{H_{ref,2}(d)}(x) \tag{3.15}$$

where

$$g_{H_{ref,2}(d)}(x) = \begin{cases} \left| I_{ref}(x) - I_2(x') \right| & if \left| I_{ref}(x) - I_2(x') \right| < Threshold \\ 0 & elsewhere \end{cases} \tag{3.16}$$

In (3.15), $N_{threshold}$ is the number of pixels that have costs below the threshold. Initial depth planes are estimated by minimizing the modified cost function in (3.15) for each segment in both of the images, as

$$L(S_i) = \arg\min_{d=1:M}(C_{S_i}(d)) \quad \forall S_i \in S \tag{3.17}$$

In other words, for each S_i in both of the views, a labeling is performed via minimization over M given depth planes. Moreover, the plane list is sorted in the ascending order of the cost values, so that the best K planes are determined for each segment. This step is also important, since the sole application plane sweeping might not give the best depth plane due to the inaccurate constant depth assumption. The determined K planes is taken into consideration in the angle sweeping step and such a limited input to the angle sweeping step speeds up the algorithm.

3.4.3.2 Angle Sweeping

In the previous step, the planes are constructed towards the direction of the principal axis of the reference camera. Therefore, all the segments are assumed to have constant depth values among their pixels. The constant depth assumption is valid as long as the size of S_i is considerably small, since small segments correspond to regions where depth variation is limited. However, as the segment size increases, the area of the surface in 3-D space becomes larger and the constant depth assumption might be violated. Actually, such a scenario is valid, as long as there are slanted or complex surfaces in the scene. Considering the planarity assumption of the proposed algorithm, only slanted surfaces are focus upon, and the complex surfaces which involve more natural shapes are simply out of the scope of this thesis.

In order to increase the range of the algorithm from the constant depth surfaces to slanted surfaces, the angle sweeping [55] is required. In angle sweeping, the segments which are assumed not to locate at constant depth (having areas larger than an *Area Threshold, AT*) are rotated in *x* and *y* directions *(*α,β*)* around their centroids (see Figure 3.10). The rotation is performed within the angle range of the viewing rays from the camera centers to the centeroid of the segment in 3-D according to the corresponding depth, since the segment should be observed by both of the cameras properly. The homographic relation between stereo images via the segment planes is not violated, since rotation only changes the normal directions of the planes. Hence, the new homographies can be determined, after accordingly modulating the plane normals with respect to the rotation angle in *x* and *y* directions as follows:

$$H_i(d) = \begin{bmatrix} P_1^i & P_2^i & P_4^i \end{bmatrix} - P_3^i \begin{bmatrix} \frac{n_1'}{n_3'} & \frac{n_2'}{n_3'} & \frac{n_4'}{n_3'} \end{bmatrix} \tag{3.18}$$

where

$$\begin{bmatrix} n_1' \\ n_2' \\ n_3' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{3.19}$$

Moreover, the rotations also do not disturb the smoothness between the pixels belonging to the same segment, since the depth values change according to the new plane parameters which defines slanted surfaces.

In angle sweeping, the cost function that is utilized in plane sweeping step is extended to a new penalty term with three parameters; two rotation angles, and a depth, to define finer descriptions of the planes as follows

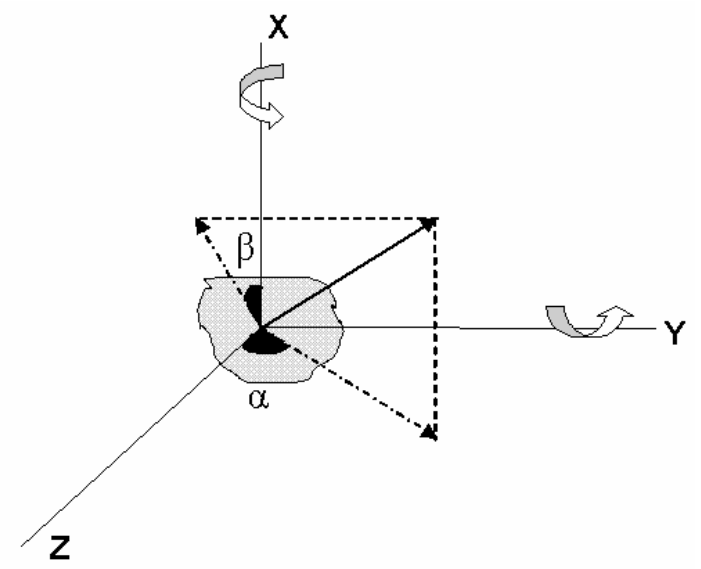


Figure 3.10: The segments are rotated around their centroids.

$$C_{S_i}(d,\alpha,\beta) = \frac{1}{N_{threshold}} \sum_{x \in S_i} g_{H_{ref,2}(d,\alpha,\beta)}(x) \tag{3.20}$$

Thus the optimization is performed in three dimensions. Moreover, the labeling function defined in (3.17) is modified according to the increase in the number of parameters to obtain, as in (3.21).

$$L'(S_i) = \arg\min_{d,\alpha,\beta}(C_{S_i}(d,\alpha,\beta)) \qquad if\ N_i > AT \tag{3.21}$$

The angle parameters for considerably small segments are set to zero. In order to save time and computation, angle sweeping is performed for only K planes which result with the best K plane sweeping costs, instead of the whole space.

As a result, after the employment of the sweeping operations consecutively for each segment in both of the stereo images; the plane parameters are extracted via angle positions (α_i, β_i) and depth locations (d_i). This information also provides the depth values for each related pixel within the segments individually. Hence, two depth maps are obtained independently for both views. The cross relation between these depth maps should be considered in order to obtain more reliable depth maps, since the extraction is performed independently up to this stage. In the next step, the depth map for the reference view is refined via left-right consistency cross check.

3.4.3.3 Consistency Checks for the Initial Depth Maps

The reliability of a depth map in stereo images is usually checked by comparing the depth maps of the two different views extracted independently [38]-[45]. However, the matching should be crosswise, in other words, if a pixel in the reference view is matched to a point in the other image by the help of 3-D information defined for one of the views, then this matched pixel should also point back the same pixel, as in Figure 3.11. In Case-1 for Figure 3.11, the estimated depth can be assumed to be reliable for the pixel in the left view, whereas for Case-2, it is obvious that the resultant depth value is erroneous. Hence, the reliable regions in a depth map can be detected by such a procedure.

In the case of segment matching, the problem, as well as the solution, is both similar. The reliable segments are detected via a left-right consistency check in pixel domain. The reliability measure of a segment is defined as the percentage of the reliable pixels that it contains. If the percentage of the reliable pixels within a segment is above a threshold (noting that **70%** is used throughout this thesis**),** then the segment can be assumed to be "reliable". After performing this check for each segment, the reliable segments and depth map can be determined.

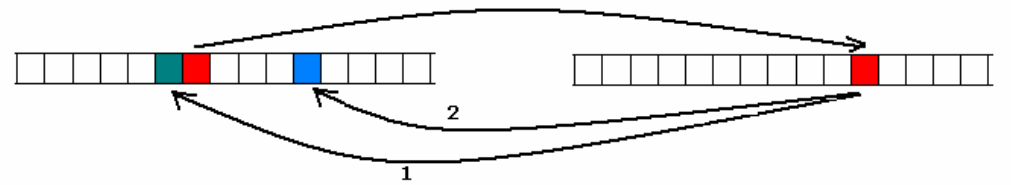


Figure 3.11: Left-right consistency check between two views

After the reliability check, there will be a set of inconsistent segments which contain unreliable pixels. The reason behind such a result can be due to noise, violation of planarity assumption, repeated structures or wrong segmentation due to lack of texture. In order to complete the resulting depth map and fill the holes due to inconsistent regions, a new search is performed via plane and angle sweeping only for the inconsistent segments. During that stage, *one-to-one correspondence property* is utilized to obtain a more realistic depth map. One-to-one correspondence is a very powerful constraint, since each pixel in the reference view should have only one correspondence in the second view, or vice-versa. In order to apply this property, initially the second image is reconstructed by (3.22).

$$I_r = \bigcup_{i=1}^{N} f_H(S_i) \qquad \textit{for each reliable } S_i \tag{3.22}$$

where R_i corresponds to the mapping of the reliable segment, S_i, in the reference view to the other view through the estimated homographies.

As a result of the reconstruction, the texture values for some regions will be undetermined, since the segments, which are not reliable, are not warped to the second image. Thus, the search for the inconsistent segments is performed within the planes which map the segments to those unfilled regions. In other words, the

unreliable segments are enforced to be mapped to the undetermined regions. During the search, the visual similarity is calculated with a similar cost function that is utilized during plane- and angle-sweeping step. The visibility can also be constrained, since the depth values for the reliable segments have been obtained. Therefore, the cost function is modified by considering the visible pixels. The updated cost function can be written as follows:

$$C_{S_i}(d,\alpha,\beta) = \frac{1}{N_{visible}} \sum_{x \in S_i} \left| I_{ref}(x) - I_2(x') \right| \text{ , } \forall S_i \text{ } unreliable \qquad (3.23)$$

where $N_{visible}$ corresponds to the number of pixels that are visible during the warping of S_i and $x' = H_{ref,2}(d,\alpha,\beta).x$

3.4.3.4 Iterative Update

In the first two steps of the proposed algorithm, over-segmentation of the stereo images is determined and then an initial depth map of the reference view is estimated via plane- and angle-sweeping. During this procedure, only the intensity similarity has been utilized and no constraints on (neighboring) segments have been considered. As mentioned at the beginning of the chapter, the depth map of a scene should be smooth, especially where the intensity variation is low. The smoothness is enforced to exist within segments by assigning planes in 3-D for each segment, but no smoothness assumption is enforced for different segments that belong to the same surface. In order to obtain smoother and more realistic depth maps, the initial estimate should be refined by some extra constraints. In this section, two more constraints are utilized for enforcing smoothness between segments and the visibility of the points.

In order to constrained the assumptions above, the minimized cost function in (3.20) is updated with a smoothness term that takes depth differences of the pixels on the boundaries of the segments, B_i's, into account, as follows

$$C'_{S_i}(D,\alpha,\beta) = \frac{1}{N_{visible}} \sum_{X \in V_i} \left| I_r(X) - I_2(X) \right| + \lambda \sum_{\substack{X \in B_i \\ X' \in NB_i}} \left| DI(X) - DI(X') \right| \quad (3.24)$$

In the above equation, DI is the current depth map, V_i is the set of visible pixels, and λ is the weighting factor of the smoothness term. In (3.24), the reconstructed image, I_r, is obtained by warping all of the segments in the reference image onto the other view via the homographies between two images as follows

$$I_r = \bigcup_{i=1}^{N} R_i \quad , \ni \; R_i = f_{H_{ref,2}(D_i,\alpha_i,\beta_i)}(S_i) \quad (3.25)$$

The reconstruction quality determines the reliability of the estimated depth map as given in (3.24). During such a reconstruction, some pixels in I_r will have more than one correspondence from the pixels on the reference image. In this case, the pixel, closer to the camera, is rendered which is actually the visibility constraint, as illustrated in Figure 3.12. The reconstructed image is stored in a *Z-buffer* [45] and the pixels on the top of this *Z-buffer* are utilized for the intensity filling in order to implement visibility. The intensity similarity between the reconstructed image and the original one is measured among the visible pixels as well.

The minimization of $C'_{S_i}(D,\alpha,\beta)$ in (3.24) with respect to D, α, β for each segment is *np-complete* and there are different types of approximate solutions, such as graph-cuts [41], belief propagation [40]. In this thesis, a method, which is

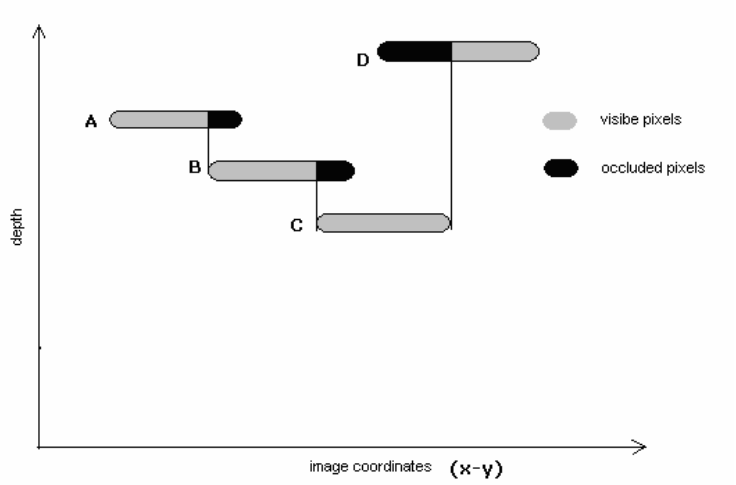


Figure 3.12: Visibility problem: Black regions indicate the invisible pixels, whereas the gray ones are visible

similar to the greedy search algorithm given in [38] and [45], is utilized. In this method, for each segment, a search is performed in the depth space bounded with the depth planes of its neighboring segments. If the segment is considerably large, angle-sweeping is also applied within the bounded region. The model, angle and depth combination, which gives the best improvement in the cost function, is assigned to the segment, if there is no more improvement; hence, the current model remains unchanged. As an example; for the segment with plane *d4* in Figure 3.13, a greedy search is performed among the depth space bounded with planes *d1*, *d2* and *d3* and the best plane model is determined. The update of these models is achieved after the search is performed for all of the segments, since the best models are determined according to the previous models of the neighboring segments. The update search is iterated until the number of the updated segments is below a threshold, (noting that 5% of the total number of the segments is used throughout this thesis).

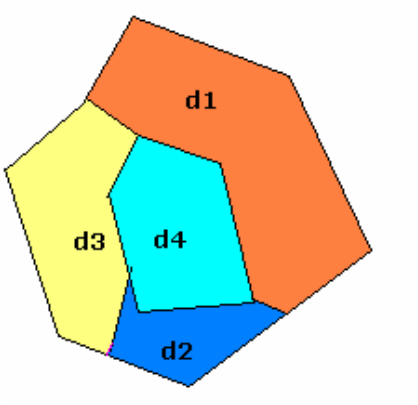


Figure 3.13: The neighboring segments bound the search space.

The iterative update method smoothes resulting depth map, since the segments are enforced to have similar planar models with their neighboring segments. Moreover, the visibility is considered rigorously and the partial occlusion problem that is explained during plane-sweeping step is also handled successfully. The contribution from the invisible pixels is neglected; hence, the intensity similarity part of the cost function becomes more reliable.

3.5 Proposed Multi-view Dense Depth Map Estimation Algorithm

If the advances in 3-D application areas are examined, popularity for multiple camera setups and multi-view video is observed in the last couple of years. The increase in the number of captured images from different cameras increases the number of the observed data for the same scene. Hence, utilizing two images, instead of multiple images in such cases makes stereo matching algorithms undesirable. As explained in the previous section, stereo matching has some problems due to the limitation of the observed data and they might fail for wider

baseline image pairs in which the scene contains occlusions frequent and repeated patterns. The utilization of multiple images removes the disadvantage of the limitation of data and makes the extension of stereo algorithms to multi-view inevitable. Considering the formulation of the proposed segment based stereo matching algorithm, the extension to multi-view is relatively simpler. The planar structures are already in 3D; hence, the projection of these planes to the additional images can be easily obtained by only the utilization of new projection matrices.

3.5.1 Extension to Multi-view Dense Depth Estimation

The general structure of the multi-view extension is also similar to the stereo matching case [56]. In the first step, the over-segmentation of the reference image takes place, as in its stereo counterpart. There is only a minor difference in the segmentation step, such that only the reference image is segmented, while the other images are not processed during this step. In the second step, the initial plane parameters for the segments in the reference view are estimated via plane and angle sweeping. Finally, the plane parameters are updated with the same optimization technique, as given in the stereo matching method. The results obtained by the multi-view extension are presented in the experimental results subsection.

3.5.2 Plane and Angle Sweeping for Multi-view Depth Estimation

Upon obtaining the candidate plane patches in the reference view, as in stereo matching algorithm, the initial plane positions and orientations are determined by the sweeping approaches. Initially, the space is divided into parallel planes among the principal direction of the reference camera. The depth values for the planes are determined according to the epipolar relations between of the reference view and

the closest neighboring view. The sweeping methods are performed sequentially, as in the stereo part. First, for each segment, the best *K* number of plane positions is determined, after minimizing a cost function based on the intensity similarity. Contribution from multiple images during warping step enhances the cost function for multi-view case over stereo matching. The mapping is provided from the reference view to the other views (*k*) as follows:

$$f_H^k : I_{ref} \to I_k \quad \ni f_H^k(S_i) = R_i^k \subset I_k$$
$$H_{ref,k}(D,\alpha,\beta)\,x = x',\, x \in S_i \text{ and } x' \in R_{i,D\alpha\beta}^k \tag{3.26}$$

In (3.26), R_i^k corresponds to a region in k^{th} image to which S_i is mapped. For each camera, the homography matrices between the reference camera and itself among depth planes (as illustrated in Figure 3.14) are determined by (3.10) and (3.11), after modulating the plane normals according to α and β as in (3.18) and (3.19). In addition, the cost function is also extended to the multiple-view case, as given in (3.27).

$$C_{S_i}(d,\alpha,\beta) = \frac{1}{\left(\sum_{\substack{k=0 \\ k\neq \mathrm{Re}\, f}}^{imageNO} N_{Threshold}^k\right)} \left(\sum_{\substack{k=0 \\ k\neq \mathrm{Re}\, f}}^{imageNO} \left(\sum_{x\in S_i} g^k_{H_{ref,k}(D,\alpha,\beta)}(x)\right)\right) \tag{3.27}$$

where

$$g^k_{H_{ref,k}(D,\alpha,\beta)}(x) = \begin{cases} \left|I_{ref}(x) - I_k(x')\right| & if \left|I_{ref}(x) - I_k(x')\right| < Threshold \\ 0 & elsewhere \end{cases} \tag{3.28}$$

In (3.27), $N^k_{Threshold}$ indicates the number of pixels of S_i that are below the threshold when the segment is warped to the k^{th} camera. Initially, the angle parameters α and β are set to zero for all of the segments in the plane sweeping

part. After the best K depth planes are determined for each segment by (3.27), angle sweeping is applied among these depth planes. Considerably large segments, where the constant depth assumption is erroneous, are rotated in x and y directions around their centroids and the best angle positions and depth locations are estimated by the minimization of (3.27). Finally, the initial depth map estimate of the reference image is obtained after the labeling of plane parameters to segments, as

$$L'(S_i) = \arg\min_{\substack{d=1:M \\ \alpha,\beta}} (C_{S_i}(d,\alpha,\beta)) \tag{3.29}$$

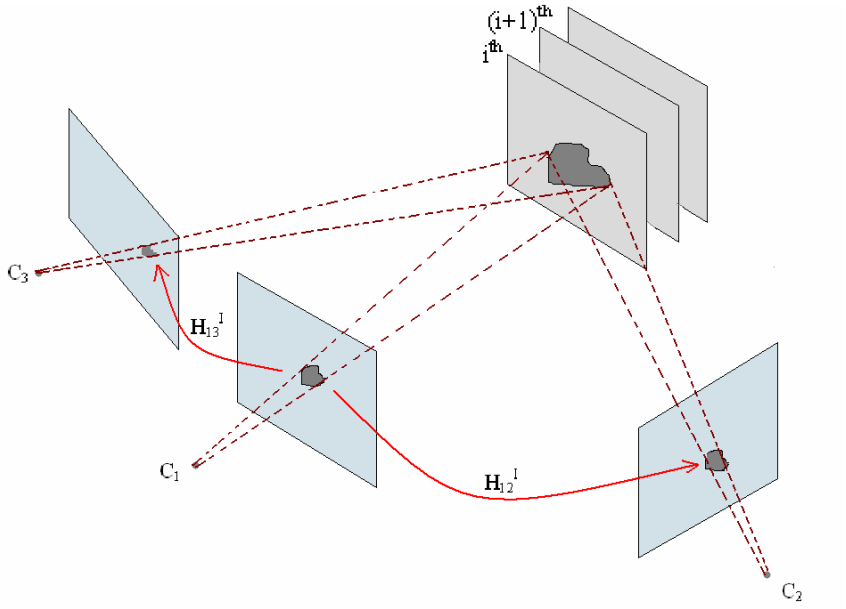


Figure 3.14: The segments in the reference view are mapped to other images via homographies.

3.5.3 Iterative Update

The initial depth map should be refined with the smoothness and visibility constraints, as explained in the stereo matching section. The method in multi-view case is similar to the update approach in stereo matching. Initially, the reconstruction of each image is performed from the depth map and the reference view as follows:

$$I_r^k = \bigcup_{i=1}^{N} R_i^k \quad , \ni \; R_i^k = f^k_{H_{ref,k}(D_i,\alpha_i,\beta_i)}(S_i) \tag{3.30}$$

Then, the cost function in (3.27) is modified with the additional smoothness term calculated among neighboring segments. In addition, the contribution of visible pixels is utilized instead of the pixels below the intensity similarity threshold in order to handle the occlusions; and the total cost is obtained by the summation of the cost values belonging to each image individually. Thus, the modified cost function can be given as

$$C_{S_i}(D,\alpha,\beta) = (L \sum_{\substack{k=0 \\ k\neq \mathrm{Ref}}}^{imageNO} \sum_{X\in V_i} \left| I_r^k(X) - I_k(X) \right|) + \lambda \sum_{\substack{X\in B_i \\ X'\in NB_i}} \left| DI(X) - DI(X') \right| \tag{3.31}$$

where

$$L = \frac{1}{(\sum_{\substack{k=0 \\ k\neq \mathrm{Ref}}}^{imageNO} N^k_{Threshold})} \tag{3.32}$$

The final depth map is estimated, after performing an iterative greedy search for each segment as in the stereo matching part. At each iteration, the models which give the best improvement in the cost function are assigned to the segments. The updates are performed at the end of the iterations and finally, the iterations stop when the updated segment number is below a threshold.

3.6 Refinement Methods

The proposed segment-based dense depth map estimation algorithm relies on the initial segmentation of the reference view. The extracted patches are treated as super-pixels and the best orientation with the depth value is searched for each super-pixel among the 3-D space. This approach provides good estimates, especially at the untextured and planar regions. Considering the general structure of the algorithm, the segmentation is based on only the color information, since there is no extra information, such as depth or any interactive segmentation as an initial input. Therefore, the reliability of the proposed method is strongly related with the initial segmentation. Hence, any errors during the segmentation of the reference view might cause errors in the estimation of the depth and orientation of the patches. Throughout the proposed method, the scene structure is assumed to be planar. In general, scenes might contain different structured objects such as curved shapes, irregular surfaces or natural shapes. The proposed algorithm is incapable of handling these cases due to planarity assumption. In order to increase the overall performance and decrease the segmentation and planarity dependency of the algorithm, some modifications are necessary.

In this part, two different solutions to the problems resulting from initial segmentation and planarity assumption are proposed, respectively. In the first sub-section, the region-split algorithm for the segmentation errors is presented. Next, the pixel-based refinement via belief propagation is introduced which is proposed to relax the planarity modeling and optimize the final depth map in pixel domain

3.6.1 Region Splitting

The region splitting algorithm aims to decrease the dependency of the initial segmentation on the depth map estimation algorithm. Due to color or intensity

similarity, some regions that are located at different depths in the 3D scene can be merged into one single segment; hence, they are forced to locate on the same plane. Such regions should be detected, separated and re-estimated to refine 3D model of the scene correctly.

Detection of mis-segmented regions is a difficult problem, since each segment has strong color consistency among their pixels. Such a property enforces to utilize additional information besides the color distribution, in order to split the mismatched segments. In the proposed approach, the extracted depth/orientation information of the segments is utilized to detect the sub-regions for splitting. The algorithm has two steps; in the first step, the segments are re-segmented within their bounds and they are divided into subregions which have stronger color consistency than the whole segment (3.33), as

$$S_i \rightarrow \{S_{i1}, S_{i2}, S_{i3}, ..., S_{im}\} \;\; \forall i \in \{1,2,..., N\} \tag{3.33}$$

After the re-segmentation step, the extracted 3D model (depth/orientation) is tested and the reliability of the model is determined for each sub-region. If $C_{S_{im}}(D_i, \alpha_i, \beta_i) > C_{S_i}(D_i, \alpha_i, \beta_i)$, then a new depth search is performed for the corresponding sub-region, S_{im}. The depth that gives the minimum cost is assigned to the region and if it is out of some predetermined bound (noting that %5 is used through this thesis) of the initial depth than the sub-region is splitted from the segment. Hence, the mis-segmented regions are detected and separated when the procedure is applied for all of the segments. As a result, a new segmentation output is obtained by the separated segments which have a stronger color consistency than the previous segments.

After the detection and splitting operations; the depth map is refined via a re-estimation of the plane parameters with the new segment distribution. During the refinement, the steps explained in stereo and multi-view matching are utilized,

which are plane-sweeping, angle-sweeping and finally, iterative update. Hence, the initial segmentation is updated with the splitted version and the final depth map is estimated.

The splitting method can be summarized in Figure 3.15. The initial segment is divided into a number of sub-regions that are illustrated with different colors and then the sub-regions that do not fit correctly to the estimated plane model of the initial segment are detected via model check and the cost value comparison. Hence, a new segment is obtained. Moreover, the separated sub-regions are also considered as distinct segments so that a finer segmentation can be obtained.

Figure 3.15: The steps of the region splitting algorithm.

3.6.2 Belief Propagation

The proposed algorithm is based on the planarity assumption, in which the scene is modeled via plane patches, and every object has planar characteristics. Such an assumption enforces each object in the scene to be composed of planes. However, this property is not valid for each case and as a result the depth map estimation can be erroneous. In order to overcome such an error, a final step is proposed to refine the depth maps in pixel wise manner. The operations performed in pixel

domain can provide us the relaxation of the estimated planar surfaces and come out with more realistic, continuous and optimized depth maps.

The belief propagation (BP) is a well known optimization technique, especially used in the Markov Random Fields and proved to work quite well [32]. BP is also widely used in computer vision discipline for dense depth map estimation algorithms and results in realistic depth maps [30] [31]. In this work, BP is utilized in order to refine depth values of the pixels, since BP can be easily adapted to the current system and have a smoothing effect that can eliminate the small discontinuities between the pixels located at the boundaries of the segments.

The method can be described as follows; the estimated depth map by the proposed planarity assumption is utilized during the initial cost evaluation of the algorithm described in [31]. In [31], cost values are assigned for the candidate depth planes for each pixel and the depth map is extracted by iterative belief propagation among the neighboring pixels. In this case, the initial costs of the pixels are evaluated according to the initial depth map as follows:

$$C_{x,y}(d_i) = |d_i - d| . \frac{1}{imageNO - 1} \sum_{\substack{k=0 \\ k \neq ref}}^{imageNO} \left| I_{ref}(x,y) - I_k(x',y') \right| \quad (3.34)$$

where d is the current depth value of the pixel (x,y) in the depth map, and $(x',y',1) = H_{ref,k}(d_i).(x,y,1)$. The depth values away from the initial depth plane of a pixel are penalized by the depth differences with the corresponding initial depth. Finally, the refined depth map is obtained by optimizing the cost function interrupted with the initial depth map.

3.7 Experimental Results

The experimental results related to the over-segmentation, plane- and angle-sweeping in stereo and multi-view dense depth map estimation algorithms are given in this section. During the experiments, different data sets are utilized in order to indicate the robustness of the proposed algorithm. The segmentation section focuses on the importance of the initial segmentation during the detection of the planar patches and two fundamental methods are compared. In the next section, the results for the individual steps for the stereo matching algorithm are presented with the extracted depth maps. Moreover, the modified cost function and its benefits over the original cost function are also investigated and clarified. Finally, the examples of multi-view dense depth map estimation and the depth map refinements are illustrated.

3.7.1 Simulations on Segmentation

The results for the over-segmentation algorithm that is performed by RSST [3] and Mean-Shift [4] algorithms on *Break-dancer* [53] multi-view image sequence are illustrated in Figure 3.16. The segmentation results are colored randomly in order to better differentiate the segments from each other. In Figure 3.17, the segmentation results for a frame from *Uli* [54] multi-view image sequence are given. In addition, the over-segmentation of the reference view of *Teddy* [52] stereo image data set is also given in Figure 3.18.

As it can be observed from Figure 3.16, Figure 3.17 and Figure 3.18, the Mean-shift algorithm segments the smooth regions and the regions that have small gradient changes better than the RSST algorithm. RSST is sensitive to small gradient changes, especially in Figure 3.16, the floor and the walls; and in Figure 3.17 the white wall are divided into multiple regions, although the regions belong

to the same structure. However, the mean-shift segmentation algorithm handles the gradient changes and detects the whole structure of similar color without dividing into multiple segments. Due to its better performance of Mean Shift algorithm, in the depth map extraction step, this algorithm is preferred in order to determine the candidate planar patches.

3.7.2 Dense Stereo Matching

In the first step of the stereo matching algorithm, the best depth planes for each segment in both of the view are determined via plane sweeping, as presented in Figure 3.19. In this figure, the lighter regions correspond to pixels closer to the reference camera, i.e., smaller depth values. The modified cost function in (3.15) and its previous version (3.14) are compared in Figure 3.20. As it can be observed, with the original cost function, the number of regions that mismatch and have unreliable depth values is higher.

At the second step of the algorithm, angle sweeping is performed in order to model larger segments and the depth maps are updated, as in Figure 3.21. After estimating the depth maps for both of the views, the reliable segments of the reference view are detected by a left-right consistency check and a new search is performed to fill the unreliable regions. In Figure 3.22, the red-colored regions indicate the unreliable segments. After the iterative greedy search optimization which considers smoothness and visibility, the final depth map is extracted and shown in Figure 3.23. The convergence curve of the algorithm, which shows the decrease in the number of refined segments along iterations, is also given in Figure 3.24. The number of the updated segments decreases sharply in the first five steps and the algorithm converges generally after 5-6 steps.

(a)

(b)

(c)

Figure 3.16: (a) Break-dancer reference image, Segmentation via (b) Mean-Shift, and (c) RSST.

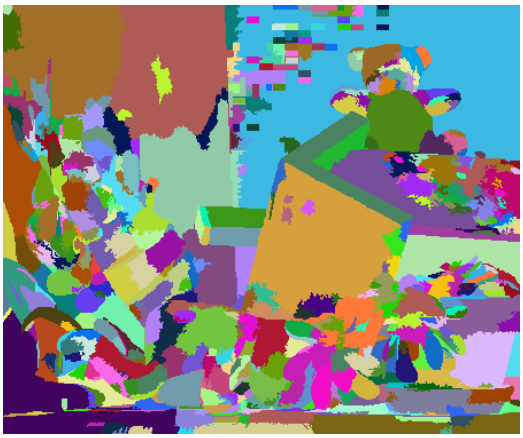
(a)

(b)

(c)

Figure 3.17: (a) *Uli* reference color image, Segmentation via (b) Mean-Shift and (c) RSST.

(a)



(b)

(c)

Figure 3.18: (a) Teddy reference image, Segmentation via (b) Mean-Shift and (c) RSST.

The algorithm is tested on the well-known stereo test bed at [52] (*Middlebury*), where many stereo dense matching algorithm results are compared with each other. The images are taken from narrow baseline cameras and the occlusions are limited. The following results have been obtained for three other images given in with their ground truths, Figure 3.25, Figure 3.26. The reconstructed and the original images are also given in Figure 3.27. The white regions in the reconstructed images correspond to the discontinuities between the segments. The proposed algorithm ranked 23rd in the overall case in which the comparison is made upon the ground truth images provided by [52]. In addition, it can be observed that the boundaries are preserved and the depth discontinuities are handled in the proposed method.

In addition to those data sets, another comparison has been performed between the proposed algorithm, an active 3D camera device which detects the depth of a scene via infrared light and a stereo (*Bumblebee*) camera. The active 3D camera is *CSEM Swissranger SR 300* and utilizes the flight time of infrared light and gives the depth field of a scene in real-time, whereas Bumblebee camera has a built-in software and estimates the depth map in real-time based on local methods. The stereo images (METU-1, METU-2) taken by the Bumblebee camera are given in Figure 3.28 and Figure 3.30; the estimated depth maps via three different methods are illustrated in Figure 3.29, Figure 3.31 accordingly. In these figures, the scales of the depth images are different for the three different methods. The results indicate that, the proposed algorithm models the planar surfaces quite well, although the texture information is not sufficient.

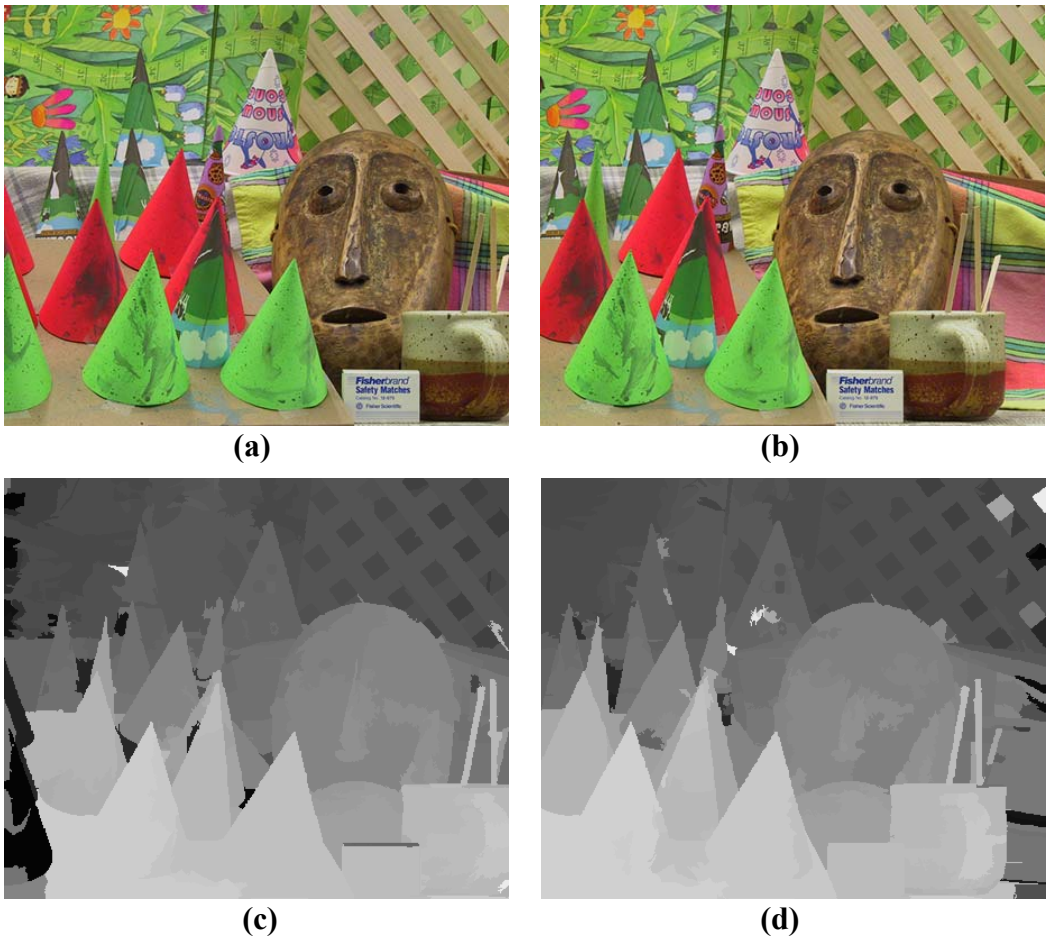


Figure 3.19: The left and right views of *Cones* [52] are given in (a) and (b) , (c) and (d) indicate the depth map via plane sweeping for both of the views.

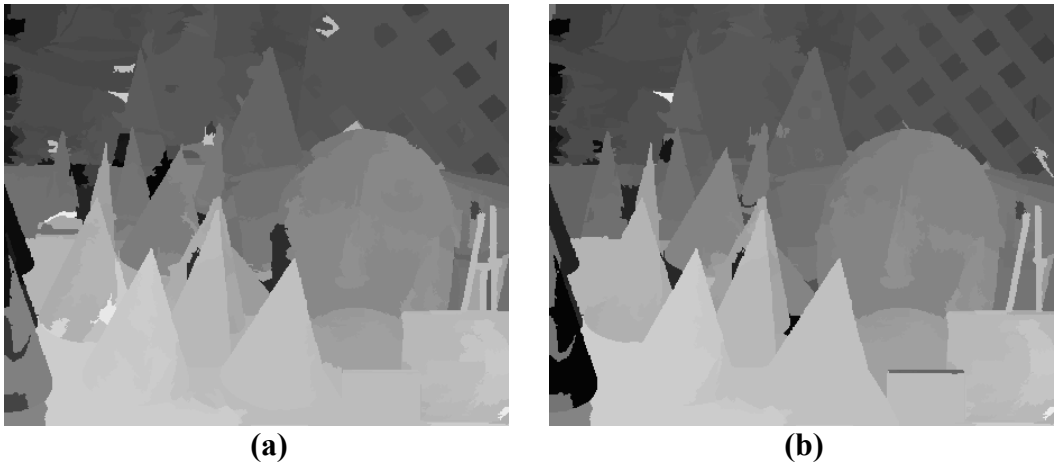
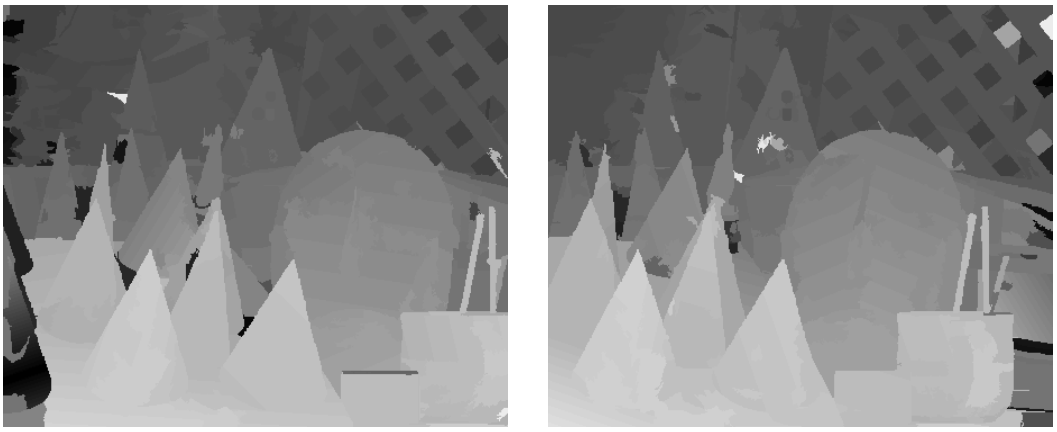


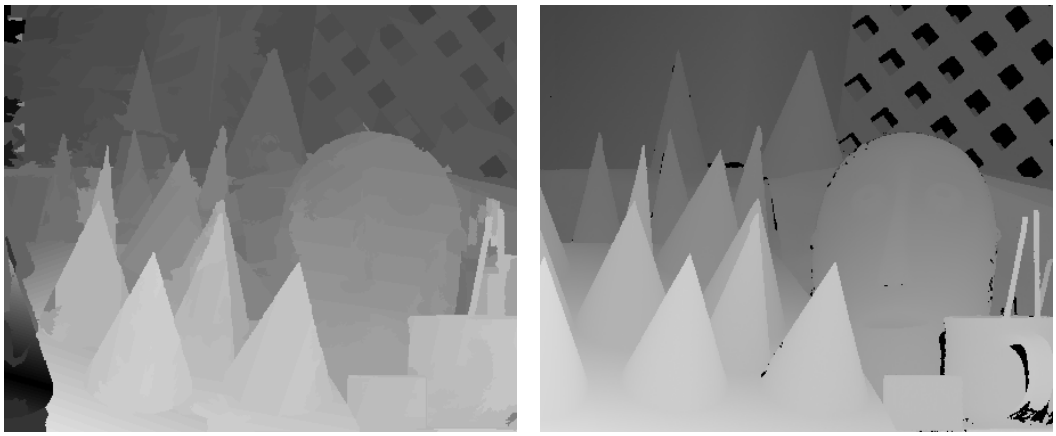
Figure 3.20: (a) The estimated depth map without thresholding in the cost function, (b) The depth map obtained after the modification.



(a) (b)

Figure 3.21: The depth maps estimated by angle sweeping for (a) left and (b) right views.

Figure 3.22: Red colored regions are detected as unreliable after the cross-check.



(a) (b)

Figure 3.23: (a) The final depth map for *Cones*, (b) the ground truth given by [52].

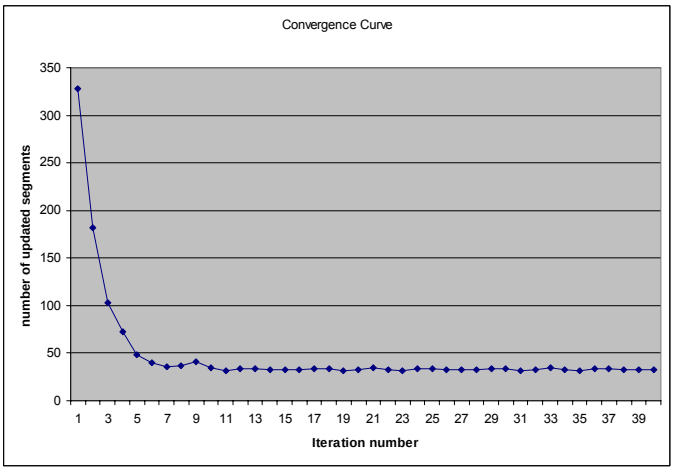


Figure 3.24: The convergence curve of the proposed iterative update algorithm

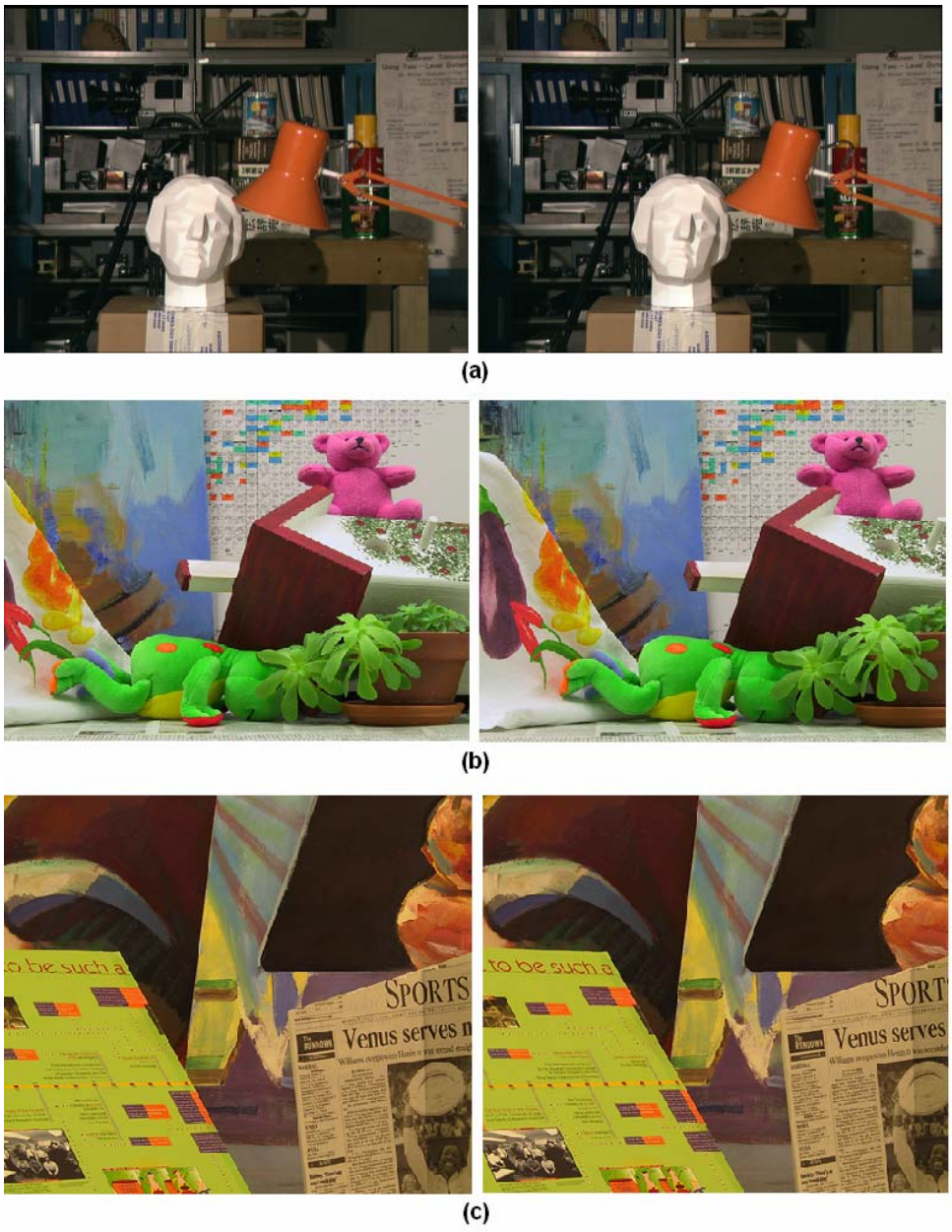


Figure 3.25: (a) Tsukuba, (b) Teddy, (c) Venus stereo data sets.

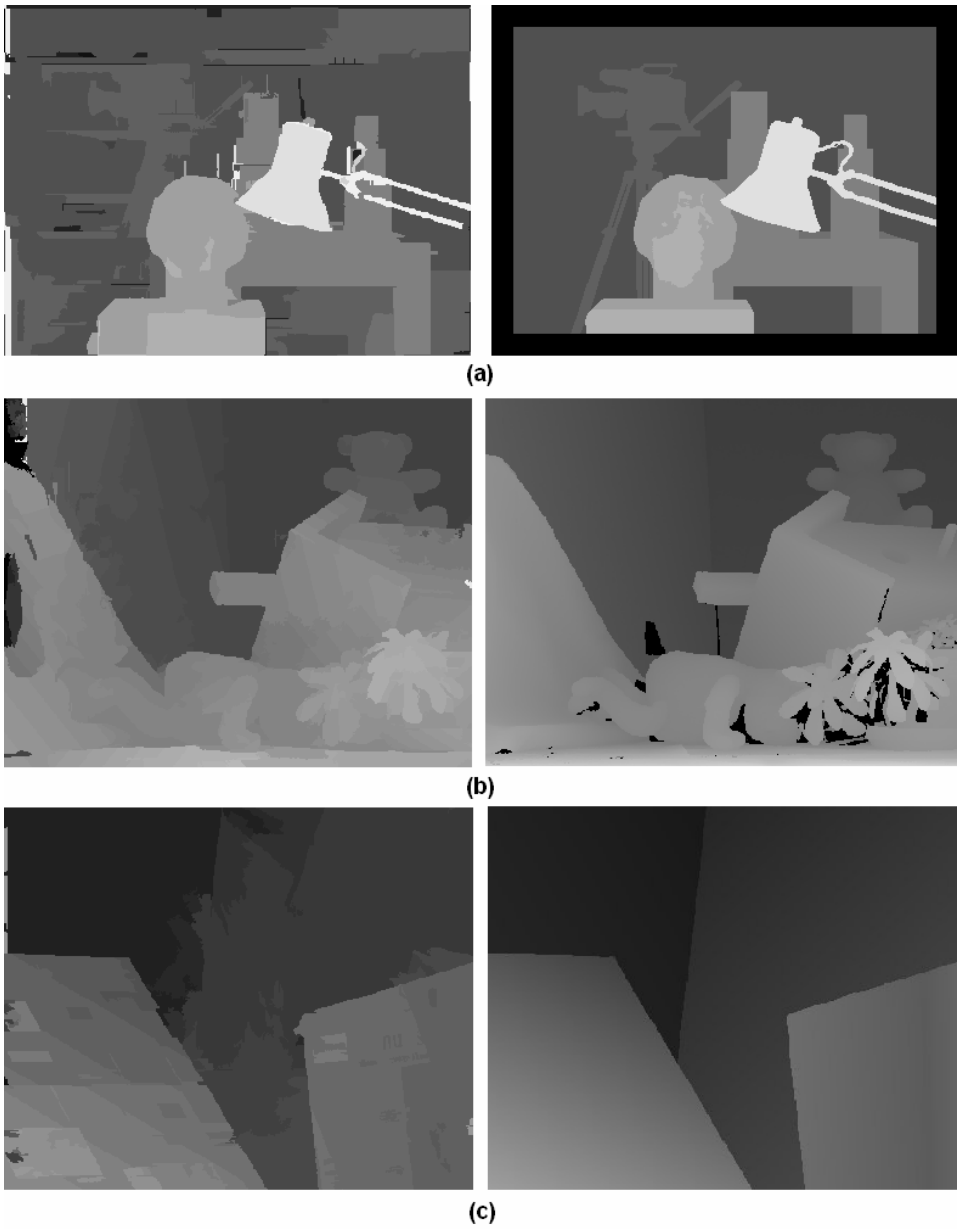


Figure 3.26: The left column is the estimated depth maps; right column is the ground truth images

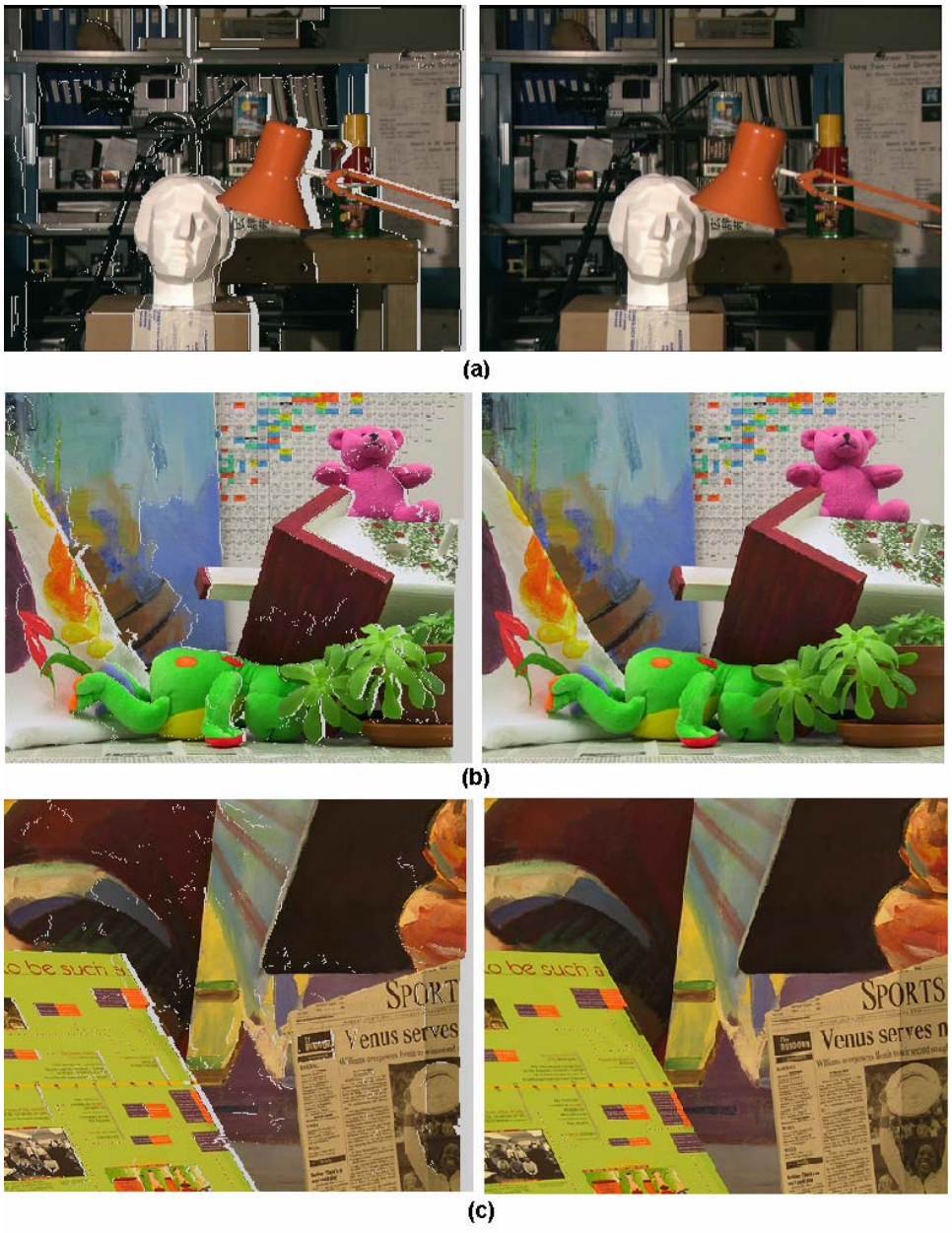


Figure 3.27: The left column corresponds to the reconstructed images; the right column is the original images.

Figure 3.28: METU-1 stereo images captured by the Bumblebee camera.

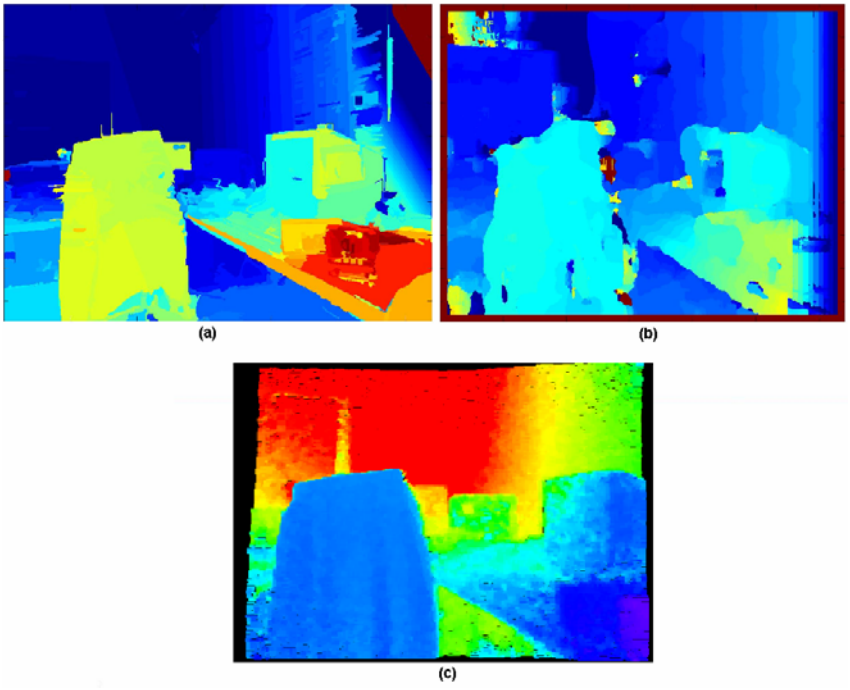


Figure 3.29: (a) Proposed method, (b) Bumblebee camera, (c) *CSEM Swissranger SR 3000* for METU-1

Figure 3.30: METU-2 stereo data that is utilized during experiments.

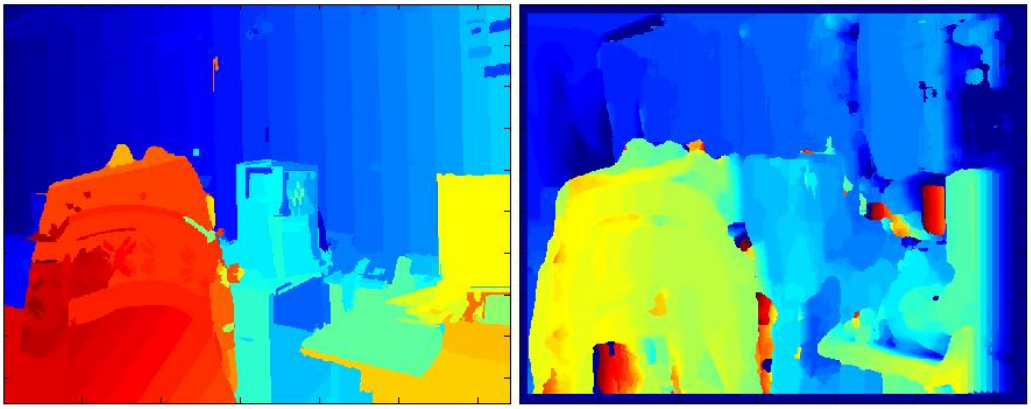


Figure 3.31: (a) Proposed method, (b) Bumblebee output for METU-2

3.7.3 Multi-view Dense Depth Map Estimation

During this part of the experiments, three different multi-view sequences are utilized, *Microsoft Break-dancer* [53], *Microsoft Ballet* [53] *and HHI Uli* [54]. The *Break-dancer* sequence is given in Figure 3.32 with the seven views from different cameras. The depth maps estimated via plane sweeping only and (plane

+ angle) sweeping are illustrated in Figure 3.33, as can be observed the (plane + angle) approach gives better results as explained previously on multi-view depth estimation chapter. The effects of the iterative update can be clearly seen in Figure 3.34. In order to compare the proposed algorithm with [46] which provides the multi-view data and the depth map is given in Figure 3.35. The depth map of [46] is smoother than the estimated depth map, but the proposed algorithm performs better [46], especially at the planar structures, such as the ground floor. The extracted 3D model is also illustrated in Figure 3.36; the ground floor is estimated without depth discontinuities as can be observed in Figure 3.36. The reconstruction of the nearest neighboring view is also illustrated in Figure 3.37 with the original view.

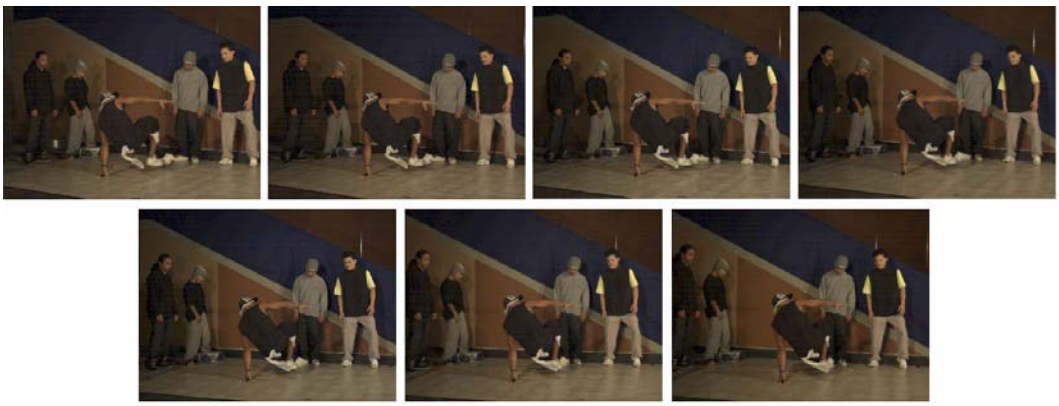


Figure 3.32: Break-dancer multi-view video sequence with seven different camera locations.

The other set of data provided by [53] is the *Ballet* sequence given in Figure 3.38 and the extracted depth field is illustrated by Figure 3.39. The planar regions and the objects boundaries are modeled reliably. *Uli* sequence is given in Figure 3.40 with the estimated depth map in Figure 3.41. In this sequence, the lightening condition changes for different cameras, however the proposed algorithm

estimates the depth map without being much affected from lightening changes due to the region matching property.

3.7.4 Refinement Methods

Some extra tests have been performed to check whether the proposed splitting method performs fine and refines the depth map after segmentation. In the first experiment setup, the reference image is intentionally segmented into small number of regions so that there will certainly exist erroneously segmented regions in the scene. In Figure 3.42, the resultant depth map of the *Cones* [52] image sequence is given with such erroneous segmentation results. The regions that do not fit to the corresponding 3D model are detected with the proposed method and illustrated as red colored regions in Figure 3.42. After the detection, the depth map is re-estimated as given in Figure 3.42. When the refined depth map is analyzed it can be observed that, some of the segmentation errors are detected and re-estimated successfully. Since the initial segmentation is obtained in small number of segments intentionally, the number of the regions to be splitted is high.

In the second experiment setup, the depth estimation algorithms are processed with a segmentation result that has sufficient number of segments, which is the typical case for the matching algorithm. Then the subregions that are mis-segmented are detected with the same method and the final depth is refined. In Figure 3.44**,** the depth maps of a frame from *Microsoft Break-dancer* (Figure 3.43) sequence are illustrated with three steps. In (a), the initial depth map is given; in (b) the mis-segmented regions are successfully detected and given in red color and in (c) refined depth map is illustrated.

(a)

(b)

Figure 3.33: (a) The depth map of 4th view estimated from 7 cameras via plane-sweeping, (b) The depth map via angle sweeping (darker regions are closer to the camera).

(a)

(b)

Figure 3.34: (a) The depth map without iterative update, (b) the final depth map.

Figure 3.35: The depth map provided by Microsoft Research [53] via [46].

Figure 3.36: The extracted 3D model of the scene.

Figure 37: (a) The reconstruction of the nearest neighboring view, (b) original view.

Figure 3.38: Ballet multi-view video sequence with five different camera locations.

Figure 3.39: Estimated depth map of 3rd view via the proposed algorithm, darker regions are closer to the reference view.

Figure 3.40: The Uli multi-view video sequence provided via [54].

Figure 3.41: The depth map of the reference view (3rd camera), lighter regions are closer to the camera.

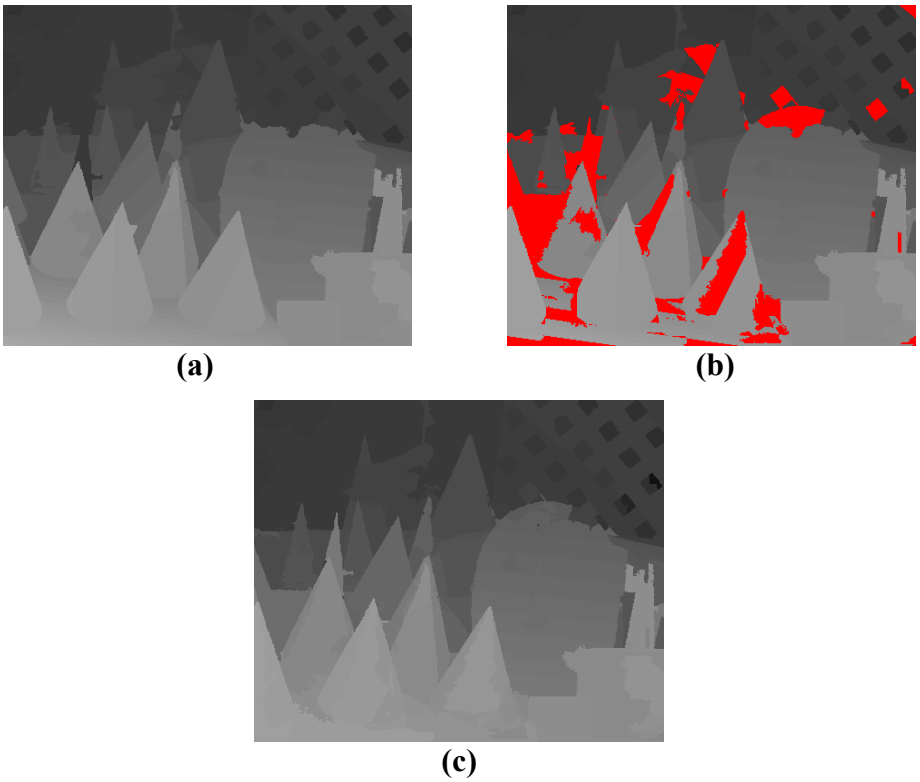


Figure 3.42: (a) The initial depth map, (b) the detected mis-segmented regions, (c) the refined depth map with new segmentation

Figure 3.43: A frame from *Break-dancer* sequence.

Finally, the effect of Belief Propagation (BP) refinement is illustrated in the following figures. The BP is applied to the estimated depth map at the final stage and the relaxation of pixels from the planar models is provided. The intensity of the depth maps are augmented in order to observe the changes clearly. In Figure 3.45, the refined depth map is illustrated. After the BP refinement, the estimated depth map becomes smoother and the mis-matched regions are handled successfully. In addition, the fuzzy effects observed at the segment boundaries are removed and a more robust depth map is obtained. Other examples corresponding to different images are given in Figure 3.46 and Figure 3.47.

(a)

(b)

(c)

Figure 3.44: (a) The initial depth map, (b) the detected mis-segmented regions, (c) the refined depth map with new segmentation

(a)

(b)

Figure 3.45: (a) The depth map with planar assumption, (b) the refined depth map via Belief Propagation.

(a)

(b)

Figure 3.46: The smoothness of the depth map increases with BP refinement over the initial depth map.

(a)

(b)

Figure 3.47: The refinement for the Uli multi-view image sequence.

CHAPTER 4

MULTI-VIEW VIDEO OBJECT SEGMENTATION

Segmentation problem is one of the most common and endeavored problems in video processing which is related with analysis, compression and visualization of the content. As mentioned in Chapter 2, segmentation can be defined as differentiating the semantically meaningful objects in a scene from each other and defining a new and meaningful representation which helps to interpret the images.

This chapter is composed of 4 parts; in the first part, a classification of video object segmentation methods and a brief literature review are presented. The following section is devoted to the static scene segmentation from multiple cameras, which includes the extracted depth information to the color within a graph structure. A weighted graph is constructed with link weights obtained according to depth and color, and then a recursive partitioning is performed in order to segment semantically meaningful regions. In the third part, dynamic scene segmentation is proposed with the update of the link weights by motion vectors. Finally, the experimental results are illustrated on various data and the comparisons between color, color and motion, and dynamic scene segmentation are given.

4.1 Literature Review for Video Object Segmentation

From the perspective of this thesis, the video object segmentation can be analyzed in two main categories, as segmentation in mono-view and multi-view video. So far, due to the limitations of the capture technology, video content was usually captured via only single camera, whose examples are commonly encountered on television or internet. Hence, most of the segmentation algorithms have been developed for only mono-view sequences, which utilize color and/or motion information among pixels. These algorithms can be classified into three main groups, as given in [57]; spatial, temporal and spatio-temporal segmentation. The spatial segmentation is performed via the color information. In spatial methods, the main assumption is about the equivalence of the boundaries of semantically meaningful objects in video with the boundaries of the similar colored patches. This assumption fails, if the foreground and background regions have smooth intensity changes in between. In addition, as soon as the objects make a relative motion between each other, even these similar colored patches display a consistency of optical flow which defines the boundary between objects. The segmentation based on the motion of the patches or pixels is defined as the temporal segmentation. However, optical flow is usually unreliable at object boundaries and might still cause mis-segmentations. In order to overcome the limitations faced during spatial and temporal segmentation, the spatio-temporal methods combining both color and motion information, have been proposed by many researchers [57].

The spatio-temporal approaches might be classified into two main categories [57]; the algorithms tracking regions from frame to frame and the algorithms that consider the whole 3D volume of the pixels within video segments. In the first category, the frames are segmented by using the motion or color information and the regions belonging to the previous frames are projected into the next frame through motion compensation; then the projections are compared with the current

regions in order to enforce the coherence between the regions [58][59][60][61]. The motion similarity is verified by the optical flow vectors. The main disadvantage of these methods is the over partition of large regions with large depth ranges, especially at the background of the image. This problem is solved with model-based approaches, such as assigning affine motion models to the pixels or patches [62] and [63]. In [63], a region-based motion segmentation method is achieved by the integration of color segmentation and motion. The second type of methods involves the segmentation of regions over the 3D spatio-temporal pixel volume obtained by the accumulation of frames into a video stack. The studies in this category have been pioneered by [64] and [65]. The methods in [66] and [67] have modeled the volume with a graph, whose nodes correspond to pixels of the consecutive frames. In this graph, the volume pixels are connected to each other via links, whose strengths are related to the feature components of color, pixel coordinates and motion vectors. The partitioning is performed by clustering the data involved in the graph.

The increase in the number of cameras provides additional information about the scene that can be extracted by two- or multiple-view geometrical relations. The 3D model of the scene can be obtained by the estimation of the dense depth field, as explained in Chapter 4. Hence, the solution to the segmentation problem in multi-view video is inevitably based on the fusion of depth and color information by the motion vectors. The multi-view segmentation has received relatively less attention so far, since the capture technology has recently begun providing multi-view data with high visual quality. The preliminary studies on this problem have started with the segmentation on stereo images, which were the only available multi-view content in the past. The depth-based segmentation is introduced by [68], which utilizes a Markovian statistical approach to obtain segmentation of the depth map estimated from the initial disparity and camera parameters. In [69], the segmentation is achieved by a depth map, refined by contour matching in which the contours are obtained from the color image. Thus, the depth and color

information are combined to refine the depth map and segment the foreground object with high precision.

The segmentation from both depth and color images can be performed in various ways. The method in [70] utilizes a Markovian framework based on shape, intensity and color cues, while [71] approaches to the problem as a displacement vector relaxation problem. In [72] and [73], the segmentation output is obtained by comparison of the edge information for both depth and color. The active contour modeling provides the conjunction of edges extracted independently from both of the images in [72]. On the other hand, a simple region growing-based segmentation that preserves the edge information is introduced in [73]. A complete system detecting the moving objects and segmenting the 3D environment, according to color, depth and motion is introduced with [74] and [75].

The research efforts on multi-view video segmentation have focused on only stereo image sequences, however as multi-view data becomes available, the segmentation problem should be considered in the context of multiple (more than two) camera scenario and the algorithms should utilize the observed data more efficiently. The requirement of the multi-view segmentation leads to extend the proposed color segmentation algorithm in Chapter 2 to the multi-view case. Such an extension could be achieved in two different ways; the stationary image segmentation which only deals with the frames at the same instant belonging to different cameras and dynamic scene segmentation involving the motion vectors in addition to color and depth.

4.2 Static Scene Segmentation from Multiple Cameras

The color segmentation of an image might contain errors, if the semantic objects have different colored sub-regions within their boundaries. An example for such a

case is given in Figure 4.1, where the body of the man has sub-regions with different dominant colors; hence, the color-based segmentation will over-partition this person into distinct sub-regions. Multi-view observation of a scene provides the essential information about the depth structure and 3D model of the scene via epipolar constraints. The approximate shape and spatial positions of the objects in the scene can be estimated by analyzing the depth or 3D information extracted form multi-views. According to the coherence principle, the regions with smooth depth variations should belong to the same object, which is a realistic assumption since common objects do not have sharp depth discontinuities. Thus, the depth information can be utilized to refine the color segmentation based on the coherence principle.

Figure 4.1: The objects may have different colored sub-regions

The proposed segmentation scheme is similar to the proposed modified normalized cut method that is introduced in Chapter 2. The image is initially over-segmented to determine the non-overlapping similar colored patches, S_i, such that each pixel in the patches belongs to the same object. Each S_i is defined by two parameters, the mean intensity of the pixels I_i, and the spatial location SL_i, corresponding to the center of mass of the patch. The parameterization of the dense depth extraction algorithm in Chapter 3 is also similar to the color segmentation parameterization. In depth map estimation, for each S_i, labeling of depth and angles is performed, as

$$L(S_i) = (D_i, \alpha_i, \beta_i) \tag{4.1}$$

in order to define the 3D structure. Hence, after the extraction of the depth map, the definition of S_i's is increased to five parameters including depth location D_i, and angle positions around x and y-axis, α_i and β_i.

A weighted graph is constructed to define similarities between nodes corresponding to S_i's, as in Chapter 2. The link weights are calculated with a function based on intensity and depth. Although, the segments are defined with five parameters; during segmentation only two of them are being utilized. The reason behind this approach can be clearly observed from Figure 4.2. During the calculation of link weights, the smoothness of the depth field between segments should be determined by the depth information only, the segments B and C in Figure 4.2 should have strong link with each other since the depth variation is smooth between them. However, if the depth planes and angle positions are utilized in order to determine the depth similarity, they will have a weak link since the centroids are located at different depths and the angle positions are diverse from each other. Thus, in order to impose the coherence principle to the color segmentation correctly, only depth values of the pixels at the segment boundaries are utilized for the depth similarity of the neighboring segments.

The similarities between non-neighboring segments, however, are detected by the depth and color parameters of the centroids of the segments, since they do not have neighboring pixels. For example in Figure 2, segments A and C will have strong link weight; however, since A is disjoint to B and B is connected with C, the effect of the corresponding link will be decreased in the recursive partitioning stage. The link weights are calculated in different ways for unconnected segments and neighboring segments consequently in (4.2) and (4.3):

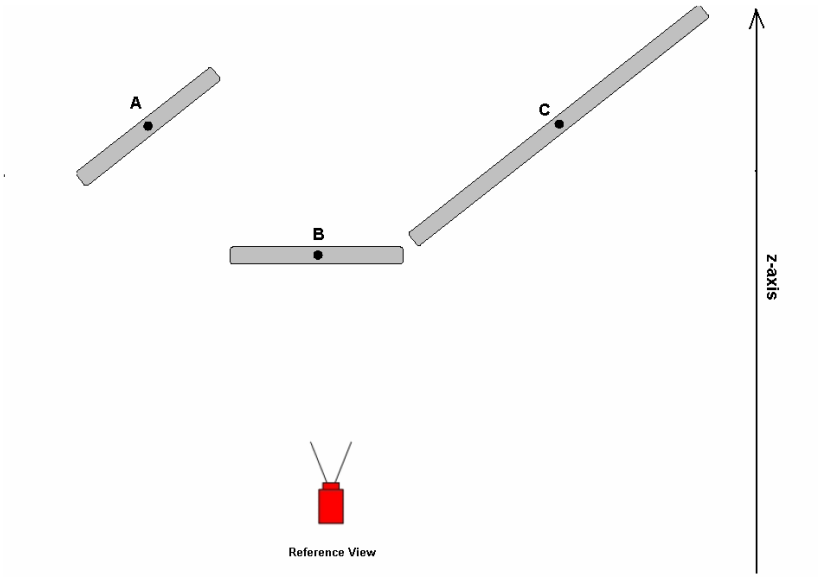


Figure 4.2: The smoothness can be imposed with depth information only.

$$w_{i,j} = \begin{cases} e^{-\left|D_i - D_j\right|_2^2 / \sigma_d} . e^{-\left|I_i - I_j\right|_2^2 / (255^2 . \sigma_i)} & \left|SL_i - SL_j\right|^2 < R \\ 0 & elsewhere \end{cases} \quad (4.2)$$

$$w_{i,j} = \begin{cases} e^{-D_{ij}^2/\sigma_d} . e^{-I_{ij}^2/(255^2.\sigma_i)} & \left|SL_i - SL_j\right|^2 < R \\ 0 & elsewhere \end{cases} \quad (4.3)$$

In (4.2), σ_d and σ_i correspond to weighting factor of the depth and color similarities, whereas in (4.3), D_{ij}^2 and I_{ij}^2 are obtained by using (4.4) and (4.5) as

$$D_{ij}^2 = \frac{1}{N_B} \sum_{\substack{x \in B_i \\ y \in B_j}} \left|D(x) - D(y)\right|^2 \quad (4.4)$$

$$I_{ij}^2 = \frac{1}{N_B} \sum_{\substack{x \in B_i \\ y \in B_j}} \left|I(x) - I(y)\right|^2 \quad (4.5)$$

where B_i and B_j correspond to boundary set of the i^{th} and j^{th} segments, D and I are the color and depth images and N_B indicates the number of boundary pixels between two segments in the equations below

The combination of depth and color is obtained by the weighting factors, σ_d and σ_i. The ratio between these factors affects the importance of the information related with the color and depth information. During the experiments, the factors are assigned such that similar contributions could be obtained form both color and depth images.

After the construction of the graph as in Figure 4.3, the same procedure given in Chapter 2 is followed during the partitioning process. In Figure 4.3, the random colored regions indicate different segments in the squared region and the nodes of the graph correspond to the centroids of the segments.

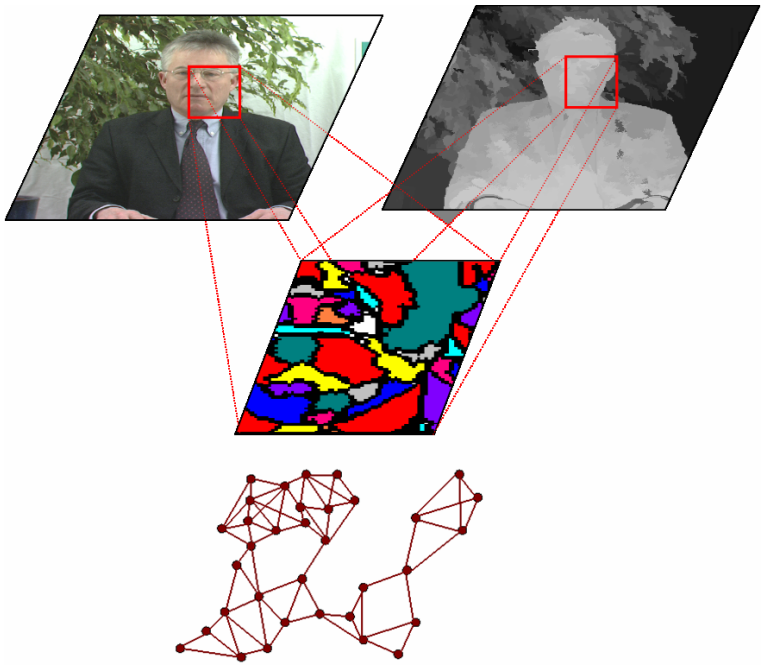


Figure 4.3: A graph is constructed based on the color and depth images, (the graph corresponding to the region in the square is illustrated).

4.3 Dynamic Scene Segmentation from Multiple Cameras

Consecutive frames of a video sequence contain motion information of the pixels with respect to time and this information is crucial for the segmentation of the moving objects in a scene. In general, it is assumed that the pixels which have similar motion vectors belong to the same object; and in the region-based segmentation case, this assumption can be stated as the regions having similar motion vectors belong to the same object. The extraction of the motion vectors from video is usually obtained by the optical flow equation. In this formulation, considering two consecutive frames with time interval t, the optical flow equation can be stated as [76]:

$$\frac{\partial I(x,y)}{\partial x}u + \frac{\partial I(x,y)}{\partial y}v + \frac{\partial I(x,y)}{dt} = 0 \tag{4.6}$$

where u and v denote the horizontal and vertical displacements of a pixel located at (x,y) whose intensity is equal to $I(x,y)$.

There are different approaches for solution of the optical flow equation. In this thesis, two different approaches are examined, which are Kanade-Lucas-Tomasi tracker (KLT) [77] and a region-based block matching method, similar to [78]. KLT assumes a constant motion vector (u,v) within a block, W, in time and spatial axis, around the center point, as shown in Figure 4.4. KLT determines the optical flow by solving the relation below derived for minimizing the error between the reference and the target regions in the consecutive frames based on the constant motion assumption.

$$\begin{bmatrix} \sum_{i,j\in W} I_x^{ij}.I_x^{ij} & \sum_{i,j\in W} I_x^{ij}.I_y^{ij} \\ \sum_{i,j\in W} I_x^{ij}.I_y^{ij} & \sum_{i,j\in W} I_y^{ij}.I_y^{ij} \end{bmatrix} \cdot \begin{bmatrix} u' \\ v' \end{bmatrix} = -\begin{bmatrix} \sum_{i,j\in W} I_t^{ij}.I_x^{ij} \\ \sum_{i,j\in W} I_t^{ij}.I_y^{ij} \end{bmatrix} \tag{4.7}$$

In (4.7), I_x and I_y indicate the derivatives of the intensity image with respect to x, y; and I_t indicate the temporal derivative (difference) between the consecutive frames. The leftmost matrix defines the cornerness of the corresponding (i,j) point. Equation (4.7) advocates that, the pixels with high cornerness values could be tracked along the frames. Thus, KLT tracker needs good feature points in the image in order to estimate optic flow.

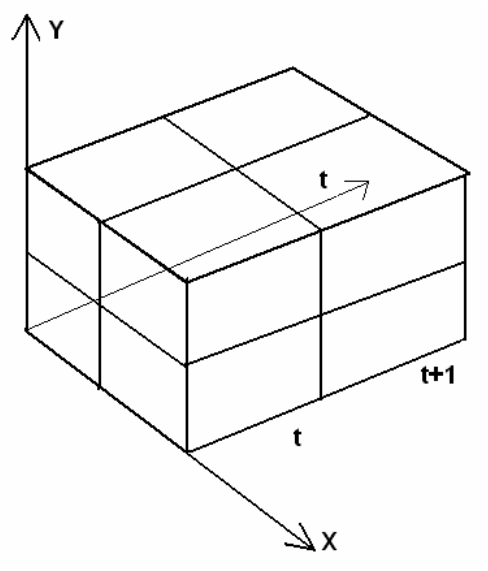


Figure 4.4: KLT assumes a constant motion vector within a window.

The aforementioned segmentation scheme utilizes segments, S_i's, in the graph structure, which are defined by depth and color. Therefore, after the estimation of the optical flow, a model is required to be assigned for each segment in order to refine the link weights with those motion vectors. The assignment for KLT method is performed by fitting an affine motion model according to the motion vectors among the pixels within each segment. The procedure is defined as follows:

1. Determine the feature points and their motion vectors via KLT tracker.
2. Within each segment check if there exist any feature points or not.
3. If there are available feature points, then fit an affine optic flow model to the segment by (4.8), given below.
4. If there is no feature point within the segment, then assign zero motion to all of the pixels in that segment.

The affine motion model can be stated as

$$\frac{a_1x+b_1y+c_1}{a_3x+b_3y+1}=u \text{ and } \frac{a_2x+b_2y+c_2}{a_3x+b_3y+1}=v \tag{4.8}$$

where the unknown parameters of the affine motion model (a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, 1) can be extracted by reformulating the problem into Ax=0 format, as below

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -v_1y_1 & -u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -u_1y_1 & -v_1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ x_i & y_i & 1 & 0 & 0 & 0 & -u_ix_i & -v_iy_i & -u_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_ix_i & -u_iy_i & -v_i \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \\ a_3 \\ b_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \tag{4.9}$$

The solution of (4.9) is obtained by performing an SVD decomposition of matrix A, and taking the eigenvector corresponding to the minimum eigenvalue as the solution.

The experiments on KLT tracker illustrate that for large moving objects and untextured surfaces in the scene, the number of feature points is not enough to assign motion models to all of the segments belonging to the object. Hence, in order to perform reliable motion assignments for the objects that involve inadequate feature points, a region-based block matching method is proposed. In this approach, constant motion model is assumed for the pixels belonging to the same segment. Thus, optical flow is estimated by an exhaustive search for each S_i

in two dimensions, x and y, such that the blocks are considered as the over-segment shapes. Finally, an iterative update of the motion vectors is performed in order to smooth the optical flows and force the neighboring segments to have similar motion vectors.

The assignment of the initial motion vectors to the segments is based on the mean absolute difference (*MAD*) similarity measure; the MAD for S_i is explicitly given in (4.10) as,

$$MAD(u,v,i) = \frac{1}{N_i} \sum_{x,y \in S_i} \left| I_1(x,y) - I_2(x+u, y+v) \right| \quad (4.10)$$

where the motion vectors for each segment are extracted by the minimization of *MAD* over the search space in two dimensions,

$$(u_i', v_i') = \arg\min_{u,v}(MAD(u,v,i)) \quad (4.11)$$

After the initial motion assignment, the motion vectors are iteratively updated with the additional smoothness constraint between the neighboring segments. In the smoothness constraint, the difference of the motion vectors between the neighboring regions are summed up and weighted in order to add to the similarity measure, *MAD* defined in (4.12). The weight of the smoothness term affects the smoothness degree of the optical flow distribution.

$$f^i(u,v) = MAD(u,v,i) + \lambda\left(\frac{1}{NB} \sum_{j \in NB_i} \left| u_i - u_j \right|^2 + \left| v_i - v_j \right|^2\right) \quad (4.12)$$

The minimization of (4.12) is obtained by an exhaustive search within the neighborhood of the initial motion vectors. The minimization procedure is given as follows:

1. For a segment S_i, that has (u_i, v_i) as initial motion vector, calculate the cost values corresponding to the motion vectors within the range of (u_i-n, u_i+n) for u and (v_i-n, v_i+n) for v component. (e.g. n is taken as 3 throughout this thesis)
2. Determine the motion vector that has the minimum cost value.
3. Perform the first two operations for every segment.
4. After all segments are visited update the motion vectors.
5. Iterate the search for several times until no updates exist.

The most important property of the region-based block matching is about the modeling of the untextured regions, such that they are modeled with larger over-segments. Moreover, during the motion assignment, the utilization of large number of pixels increases the reliability of the model. In addition, the size of the segments changes according to the texture distribution of the image. As a result, high textured regions are represented with more segments, whereas untextured regions are modeled with less number of segments. This property provides adaptive detail scale for the representation of the regions according to the region complexity.

The dynamic scene segmentation is achieved by the additional motion information after the estimation of the optical flow. The graph links are updated with the motion vectors as follows:

$$w'_{i,j} = \begin{cases} w_{i,j} \cdot e^{-(|u_i - u_j|^2 + |v_i, v_j|^2)/\sigma_m} & |SL_i - SL_j|^2 < R \\ 0 & elsewhere \end{cases} \quad (4.13)$$

where σ_m is the weighting factor of the motion vectors.

4.4 Experimental Results

The experiments are divided into two parts; in the first part, the effect of 3-D depth information on the segmentation of the image is analyzed. During these experiments, the images that are obtained from stereo and multiple cameras are utilized in order to extract the depth information. The second part involves the additional temporal information of the video sequences, belonging to the reference camera and the segmentation is further refined via the optical flow information of the objects in the scene.

4.4.1 Static Scene Segmentation from Multiple Cameras

Uli multi-view sequence, in Figure 4.5, is utilized as the first test data with its depth map obtained by the algorithm explained in Chapter 3. The segmentation is performed for only color, only depth and color with depth images. During the partitioning, the color weighting factor, σ_i, is taken as 0.0005 and depth weighting factor σ_d is taken as 0.003. The segmentation results for the random colored regions are given in the Figures 4.6, 4.7 and 4.8.

(a) **(b)**

Figure 4.5: The color image and estimated depth map for a time instant from *Uli* multi-view image sequence.

Figure 4.6: Segmentation via only color information.

Figure 4.7: Segmentation via only depth information.

Figure 4.8: The segmentation with depth and color.

The increase in the performance of the segmentation can be observed easily; the semantic objects are extracted better (such as head and body), instead of classifying the image as the composition of foreground and background. *Ballet* is the second multi-view image sequence, shown in Figure 4.9, whose segmentation results are presented in Figure 4.10. As it can be observed, the (color + depth) segmentation gives more reliable representation of the scene with respect to using only color segmentation. In these examples, the objects are segmented successfully based on only the depth information, since the objects are located at different depths.

In Figure 4.11, a frame from *Akko&Kayo* multiple image sequence [79] and the depth map estimate are given. The color and (color + depth) segmentation results are given in Figure 4.12 for the corresponding image. The refinement of the segmentation with the additional depth information is clearly observed.

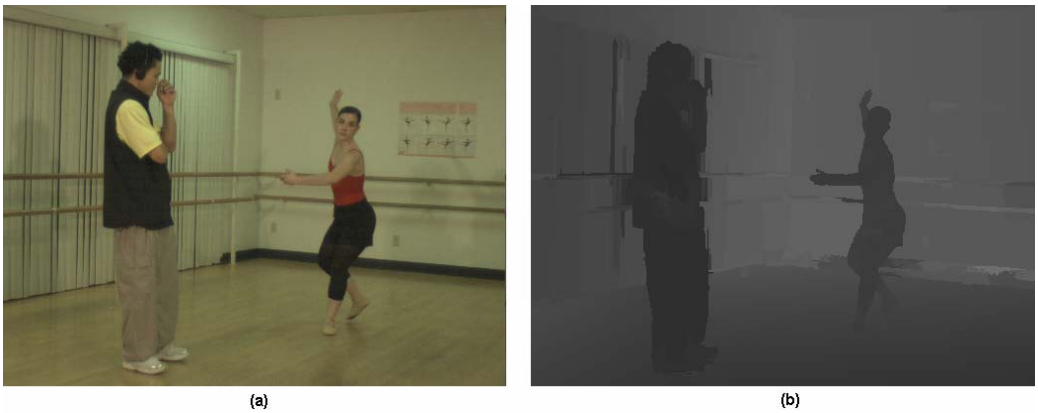
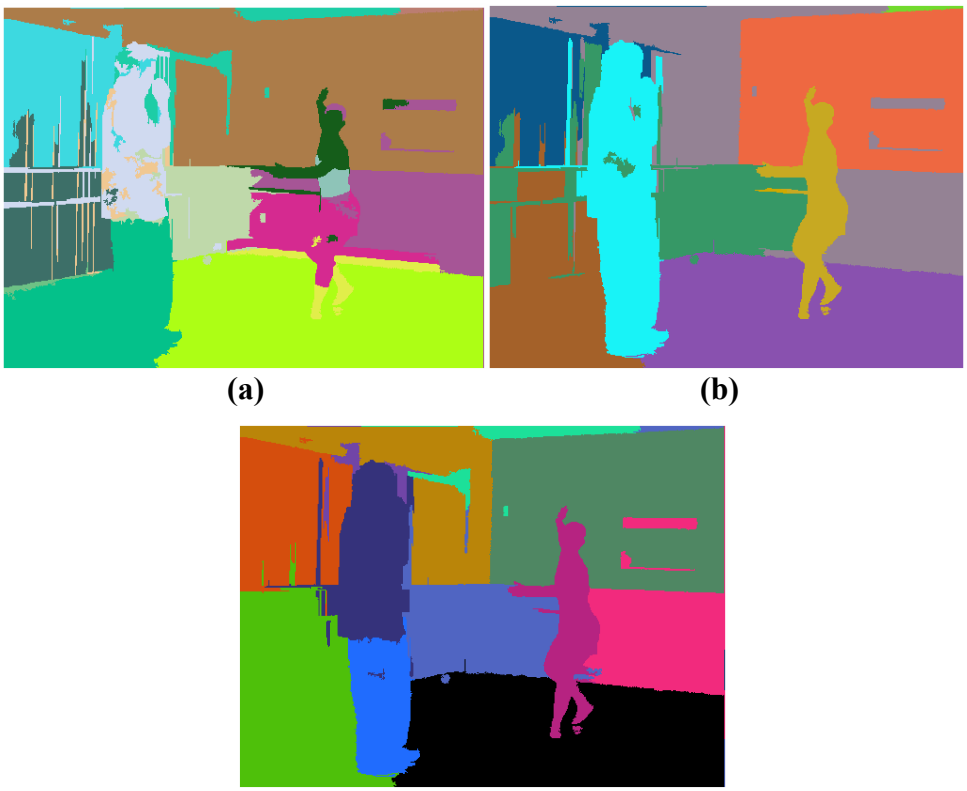


Figure 4.9: A frame of the reference view from *Ballet* sequence with the estimated depth map.



(a) (b)

(c)

Figure 4.10: (a) the segmentation with only color, (b) segmentation with only depth, (c) the depth and color segmentation.

(a) (b)

Figure 4.11: A frame from the *Akko&Kayo* sequence and its estimated depth map

Figure 4.12: (a) segmentation with only color, (b) segmentation both with depth and color.

4.4.2 Dynamic Scene Segmentation from Multiple Cameras

During the experiments of this part, the consecutive frames of the reference view are utilized in order to extract the optical flow. Initially, the optical flow estimates of KLT tracker and proposed region-based block matching method are compared. The first image sequence is from *Akko&Kayo* multiple image sequence [79] image sequence [79], as shown in Figure 4.13. The estimated motion vectors are illustrated in Figure 4.14 with their magnitudes, where blue colors indicate the stationary regions. The region-based block matching preserves the object boundaries and estimates the motion vectors of the untextured regions robustly. Moreover, KLT tracker estimates the motion vectors of the textured and high cornerness regions quite acceptable. However, KLT tracker fails at the object boundaries and stationary uniform colored regions. As a result, in the motion segmentation, region-based block matching is utilized due to its reliable motion estimates. The estimated depth map of the first frame is given in Figure 4.15, whereas the segmentation results obtained with (color + depth); and (color + depth + motion) are illustrated in Figure 4.16. The refinement after the inclusion

of the motion information can be observed clearly, since the motion vectors improve the information of depth and color and detect the moving objects in the scene. Finally, in Figure 4.17 the segmentation results of the frame in Figure 4.10, are illustrated with the additional motion information. The estimated motion vectors of the corresponding frame are also given in Figure 4.18. The effect of the motion information on segmentation quality is clearly observed; the boundaries between the moving objects and stationary regions are extracted very well.

Figure 4.13:Two consecutive frames of the reference camera of *Akko&Kayo multi-view video*

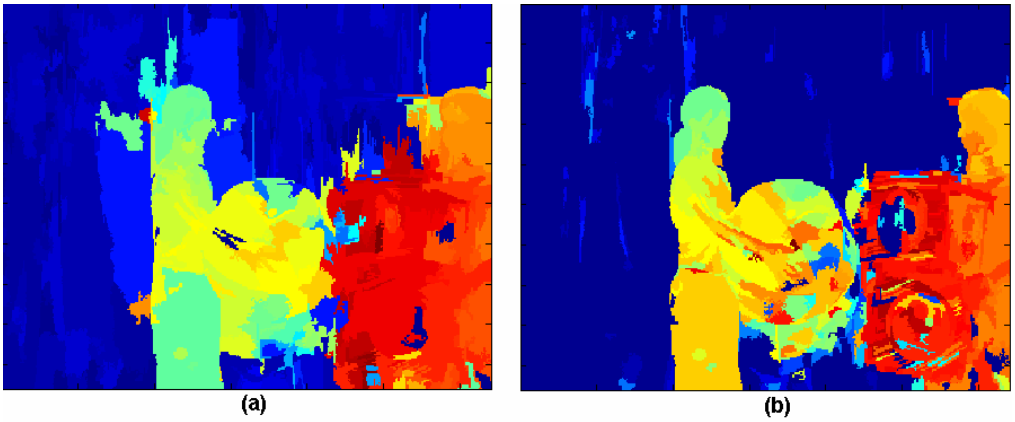


Figure 4.14: The estimated motion vectors, (a) KLT, (b) Region-based block matching

Figure 4.15: The depth map of the first frame.

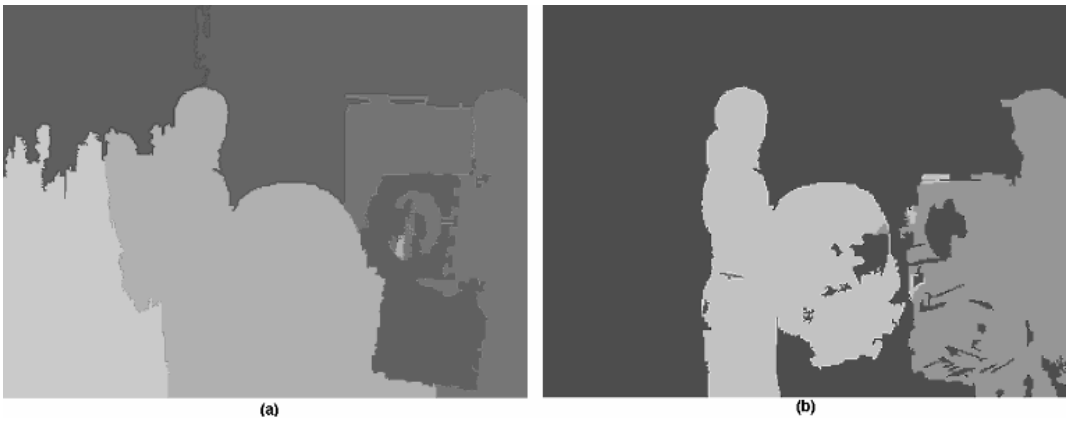


Figure 4.16: (a) (color + depth) segmentation, (b) (color + depth + motion) segmentation results.

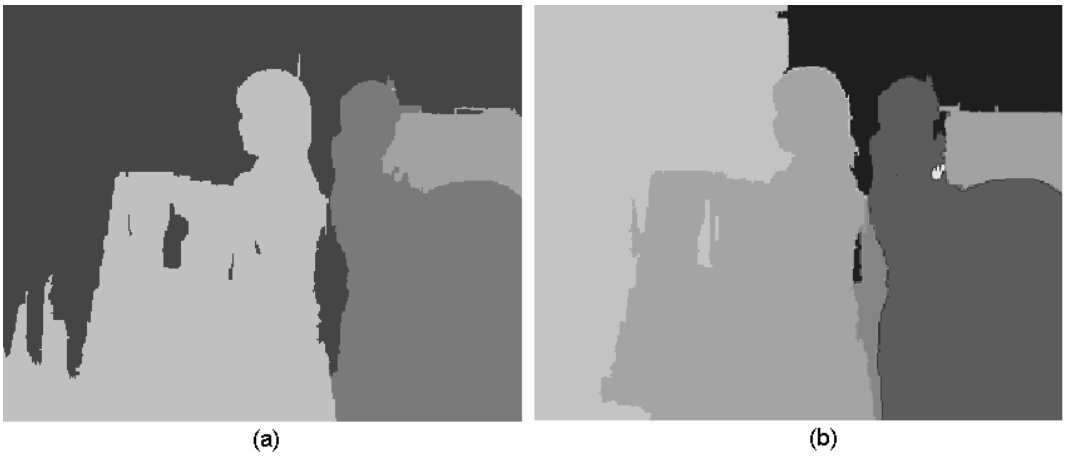


Figure 4.17: (a) (color + depth) segmentation, (b) (color + depth + motion) segmentation results.

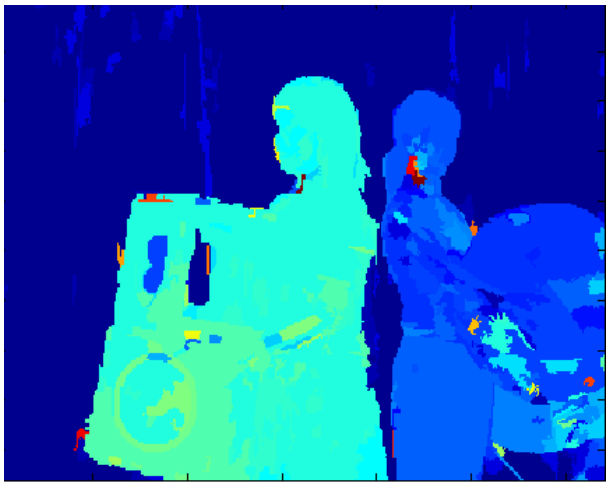


Figure 4.18: The magnitude of the estimated motion vectors, lighter regions indicates larger motion.

CHAPTER 5

CONCLUSION

5.1 Summary of the Thesis

In this thesis, novel methods for dense depth field estimation and object segmentation from mono, stereo and multiple views are presented. As the first contribution, a graph-theoretic color segmentation algorithm is proposed. This method improves the well-known Normalized Cuts segmentation algorithm with some modifications on its graph structure. Small-sized regions that are obtained after over-segmentation of the image are utilized, as the nodes of the weighted graph, instead of the conventional utilization of image pixels. The color similarities between these subregions are utilized in order to calculate the link weights between the nodes (regions). Segmentation is achieved by partitioning the graph through minimization of the normalized cuts measure. Moreover, for a region-based approach, a bias, due to varying node distribution (note that there is constant number of nodes for pixel-based strategy due to the regular grid) on the normalized cut measure is also improved by using some limitations on the number of links for each node. The proposed segmentation scheme is compared with some well-known segmentation methods, such as Recursive Shortest Spanning Tree and Mean-Shift and the conventional Normalized Cuts. The simulation results on different type of images show clear improvements over these traditional methods.

The proposed region-based approach is also utilized during the dense depth map estimation step, based on a novel plane- and angle-sweeping strategy. The regions are considered as super-pixels and the whole scene is assumed to be region-wise planar, in which the super-pixels correspond to these planar patches. The position and rotation of the plane patches are estimated robustly by minimizing a segment-based cost function, which considers occlusions as well. The quality of depth map estimates is measured with reconstruction quality of the conjugate views, after warping segments into these views by the resulting homographies. Then, a greedy-search algorithm is applied to refine the reconstruction quality and update the plane equations with visibility constraint. In the final step, two refinement techniques, region splitting and pixel-based Belief Propagation are proposed in order to refine the depth maps and relax the planarity assumption of the scene. The algorithm is applied on different stereo and multi-view data sets that result with high quality depth maps and indicate the robustness of the method among different type of scenes.

Finally, a novel multi-view video object segmentation is presented, as a result of an extension of the proposed image segmentation algorithm by updating the link strengths with the additional depth and optical flow information. In the first step, segmentation of a scene from multiple cameras is obtained and the refinement of the segmentation results over color segmentation is observed with the additional depth information. In the following step, optical flow constrained is incorporated in addition to depth and color for segmentation of the objects. The optical flow estimation is obtained via two different methods, which are KLT tracker and region-based block matching and comparisons between these methods are performed. During simulations, the improvement in video object segmentation is clearly observed, as a result of the utilization of structure and motion information in addition to color.

5.2 Discussions

During simulations, it is observed that the top-to-down characteristic of the Normalized Cuts image segmentation algorithm might cause the local color distribution to be ineffective in the graph, since each node corresponds to pixel intensities and the number of the links is extremely high. In such a complex graph structure, the global link weights become more important than the individual links. However, utilization of over-segmented regions, as the graph nodes, provides an initial step of combining similar colored pixels in the local area. Hence, the details of the image are modeled effectively in the graph. Therefore, the final segmentation with the proposed method results in more detailed partitioning of the image than the traditional Normalized Cuts.

The other fundamental drawback of the Normalized Cuts method are the requirement of large memory and the slow operation speed, which are the results of intense graph structure. In the proposed modification scheme, the decrease in the number of nodes also decreases the complexity of the graph and the segmentation is achieved faster with less memory. However, utilization of high number of segments causes varying (not constant) node distribution in the graph. Such a distribution also creates a bias towards the nodes having high number of links and the importance of link weights is decreased. In order to overcome such a bias, the number of links for each node is made limited and a more uniform distribution of the links among the nodes is tried to be achieved. The limitation increases the segmentation performance, as presented in the results of the experiments.

The region-based approach for the dense depth map estimation provides object boundaries and depth discontinuities to be preserved at the estimated depth maps, since discontinuities and object boundaries are inline with that of over-

segmented region boundaries, in general. Moreover, the untextured regions are modeled by large regions which include high number of pixels. Hence, as a result, the depth maps of these regions are estimated quite accurately in terms of their boundaries, whereas pixel-based approaches fail, especially at untextured regions. However, the initial over segmentation affects the success of the algorithm, since the planar regions are obtained by the over-segmented regions. If the pixels located at different depths are grouped into the same segments, then the resultant depth map might be erroneous. In order to overcome such cases, region splitting refinement is proposed as a final step and the regions, which are segmented incorrectly, are detected.

The multiple camera extension for the dense depth map estimation algorithm is achieved in a relatively straightforward manner, due to the fact that 3D planes are utilized during plane- and angle-sweeping methods. The increase in the number of cameras yields more reliable estimation of the depth map. Moreover, in addition to color similarity, the visibility and smoothness constraints also improve the estimated 3D models during the greedy search optimization. The dependency of the proposed algorithm on the initial segmentation is decreased by region splitting. The region split algorithm detects the incorrectly segmented regions, and refines the depth map at the final step. However, the detection of all of the mis-segmented is not achieved, since the strong color consistencies within the pixels of the segments prevent any the splitting of the subregions. The planarity assumption of the scene is not valid, as long as there are curved and more natural shapes, such as trees or balls. Thus, the pixel-based BP refinement is proposed that provides smoother depth maps and removes the fuzziness at the object boundaries.

The segmentation performance is increased with the additional depth information for the multiple camera scenario. During the experiments, the

weight coefficients of the links related with color and depth are selected to be close to each other, since the importance between these different information sources is not known a priori. The segmentation method is extended to multi-view video object segmentation by additional motion information. The motion vectors are determined from the video sequence of the reference view in the multiple camera setup for two different ways. In the first method, KLT tracker determines the feature points and their motion vectors between two frames. However, KLT tracker fails in the flat regions and around actual object boundaries, thus the motion assignment to the segments is not achieved robustly. In order to assign robust motion models for each segment, a region-based block matching is utilized. According to the experimental results; the object boundaries, flat regions and textured surfaces are handled by the proposed method. The comparisons between KLT and region-based block matching show that, region-based block matching gives more reliable results for the graph-theoretic video object segmentation purpose.

5.3 Future Work

Dense depth map estimation from stereo or multiple cameras is achieved for only one view which is considered as the reference view; however, 3D applications, such as 3DTV, might require depth field for each view. Thus, the proposed region-based approach might be extended in order to estimate the depth field, not only for a single view, but for all of the cameras. The extension might provide more reliable depth maps, as long as the cross-checks are performed between the neighboring cameras. In addition, different global optimization techniques, such as graph cuts or belief propagation, might be applied for the assignment of the depth planes to segments instead of greedy search algorithm in order to improve the depth map quality.

REFERENCES

[1] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", *John Wiley&Sons, New-York, 1973.*

[2] K S. Fu and J.K. Mui, "A Survey on Image Segmentation" *Pattern Recognition, Vol. 13, pp. 3-16, 1981.*

[3] Ozan Ersoy, "Image Segmentation with Improved Region Modelling" *Master of Science Thesis, METU 2004.*

[4] C. Dorin, M.Peter , "Mean Shift : A robust approach toward feature space analysis", *IEEE Transactions on pattern analysis and machine intelligence vol:24,no:5, May 2002*.

[5] http://robots.stanford.edu/cs223b/index.html, last accessed: 25.07.2007.

[6] J.Shi and J.Malik "Normalized cuts and image segmentation" *Proceedings of IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition, Jun 1997.*

[7] M.Stoer and F.Wagner, "A simple minimum cut" *Algorithms –ESA'94 pages 141-147, 1994.*

[8] S.Sarkar and P.Soundararajan,"Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata", *IEEE*

Transactions on Pattern Analysis and Machine Intelligence,22(5):504- 525, May 2000.

[9] Timothee Cour, Florence Benezit, Jianbo Shi, "Spectral Segmenatation with Multiscale Graph Decompositon" *IEEE International Conference on Computer Vision and Pattern Recognition, 2005.*

[10] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences", *IEEE Transactions, Circuit and Systems for Video Technology, Vol. 10, 2000.*

[11] S. H. Kwok, A. G. Constantinides and W. Siu, "An Efficient Recursive Shortest Spanning Tree Algorithm Using Linking Properties", *IEEE Transactions, Circuit and Systems for Video Technology, Vol. 14, 2004.*

[12] W. Yair, "Segmentation using eigenvectors: a unifying view" *In Proceedings of the International Conferance on Computer Vision, volume 2, pages 975-982,1999*

[13] Sarkar and P.Soundararajan,"Analysis of MinCut,Average Cut and Normalized Cut Meaures", *Extended Abstract*

[14] G.H. Golub and C.F. Van Loan, "Matrix Computations". *John Hopkins press, 1989*

[15] D. Nister, "Automatic passive recovery of 3D from images and video", *Proceedings of 2nd International Symposium on 3D Data Processing, Visualisation and Transmission, Vol.00, pp. 438-445, 2004.*

[16] R. Hartley, A. Zisserman, "Multiple view geometry", *Cambridge University Press, UK, 2003.*

[17] Fred Rothganger et'al "3D object modeling and recognition using affine-invariant image descriptors and multi-view spatial constraints" *International Journal of Computer Vision, volume 66, Issue 3, pg 231-259, 2006.*

[18] Kang S.B. and Shum H.Y. "A Review of image-based rendering techniques" *International Conference on Computer Graphics and Interactive Techniques, pg 107-116, 2001.*

[19] Ayache, N. Hansen, C. "Rectification of images for binocular and trinocular stereovision" *International Conference on Pattern Recognition, volume 1, pg 11-16, 1998*

[20] D. Sharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". *In International Journal of Computer Vision, Volume 47, pg 7-42, April 2002.*

[21] M.Okutomi and T.Kanade, "A multiple-baseline stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence, April 1993.*

[22] S.Christopp et al," Dense matching of multiple wide-baseline views", *in International Conference on Computer Vision 2003.*

[23] S.B.Kang *et al, "*Handling occlusions in multi-view dense stereo*", in Computer Vision and Pattern Recognition conference 2001*.

[24] V.Kolmogorov and R.Zabih,"Multi-camera scene reconstruction via graph cuts*". In International Conference on Computer Vision, volume II, pages 508-515, 2001.*

[25] K.Kutulakos and S.Seitz,"A Theory of Shape by Space Carving", *International Journal of Computer Vision*, 38(3):197-216, 2000.

[26] Takeo Kanade and M. Okutomi "A stereo matching algorithm with an adaptive window: Theory and experiment" *IEEE Transactions on Pattern Analysis andMachine Intelligence, 16(9):920:932.*

[27] Daniel Scharstein and Richard Szeliski "Stereo matching with nonlinear diffusion" *International Journal of Computer Vision, 28(2):155-174, 1998*.

[28] Y.Boykov, O.Veksler and R.Zabih "A variable window approach to early vision" *IEEE Transactions of Pattern Analysis and Machine Intelligence, 20(12):1283-1294, 1998.*

[29] M. Okutomi and T. Kanade "A locally adaptive window for signal matching" *International Journal of Computer Vision, 7(2):143-162, 1992.*

[30] Pedro F. Felzenszwalb and Daniel P. Huttenlocher "Efficient belief propagation for early vision". *International Journal of Computer Vision, Vol. 70, No. 1, October 2006.*

[31] J. Sun, N.N. Zheng, and H.Y. Shum "Stereo matching using belief propagation" *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 25(7) Pages: 787-800 200*

[32] Jonathan S. Yedidia, et.al "Understanding belief propagation and its generalizations" *Exploring Artificial Intelligence in the New Millennium, ISBN 1558608117, Chap. 8, pp. 239-236, January 2003 (Science & Technology Books)*

[33] Y. Boykov et.al "Fast approximate energy minimization via graph cuts" *IEEE Transactions of Pattern Analysis and Machine Intelligence 23(11):1222-1239, 2001.*

[34] V. Kolmogorov and R.Zabih "Computing visual correspondence with occlusions using graph cuts" *In International Conference on Computer Vision, volume 2, pages 508- 515, 2001.*

[35] O. Veksler "Efficient graph-based energy minimization methods in Computer Vision" *PhD thesis, Cornell University, 1999.*

[36] G. Van Meerbergen, M. Vergauven, M.Pollefeys, and Luc Van Gool, "A hierarchical symetric stereo algorithm using dynamic programming*", In International Journal of Computer Vision 47(1/2/3), 275- 85, 2002*

[37] C. Lei, J. Selzer, and Y. Yang. "Region-tree based stereo using dynamic programming *optimization" IEEE Conference on Computer Vision and Pattern Recognition 2006.*

[38] M. Bleyer and M. Gelautz. "A layered stereo algorithm using image segmentation and global visibility constraints". *In International Society of Photogrammetry & Remote Sensing Journal 59 (2005) 128– 150*

[39] S.Birchfield and C.Tomasi, "Multi-way cut stereo for stereo and motion with slanted surfaces" *in International Conference on Computer Vision 1999.*

[40] A. Klaus et all. "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure*"* In *International Conference on Pattern Recognition 2006.*

[41] Sang Yoon Park, Sang Hwa Li, and Nam Ik Cho, "Segmentation based disparity estimation using color and depth information*", In International Conference on Computer Vision 2004*

[42] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts*"*, *in IEEE Conference on Computer Vision and Pattern Recognition, 1, pp. 74-81, 2004.*

[43] Michael H.Lin "Surfaces with occlusions from layered stereo" *PhD Thesis, Stanford University 2002.*

[44] C.L. Zitnick and T. Kanade "A cooperative algorithm for stereo mathcing and occlusion detection" *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 7, July, 2000, pp. 675 - 684.*

[45] H. Tao and H.S. Sawhney "Global matching criterion and color segmentation based stereo" *in Applications of Computer Vision, 2000, Fifth IEEE Workshop pages:246-253.*

[46] C.L. Zitnick et.al "High quality video view interpolation using a layered representation" *In ACM SIGGRAPH and ACM. Trans. on Graphics, volume 23(3), pages 600–608, 2004.*

[47] Y.Wei and L.Quan "Region based progressive stereo matching" *in IEEE Conference on Computer Vision and Pattern Recognition 2004.*

[48] M. Bleyer and M. Gelautz. "Graph-based surface reconstruction from stereo pairs using image segmentation". *Proc. SPIE, vol. 5665, January 2005.*

[49] Y.Zhang and C.Kambhamettu “Stereo matching with segmentation-based cooperation” *in Proceedings of the 7th European Conference on Computer VisionPages: 556 – 571, 2002.*

[50] C.Cigla, X.Zabulis and A.A. Alatan “Segment-based stereo matching via angle and plane sweeping” *in 3DTV Conference, May 2007.*

[51] Robert T.Collins, “A space-sweep approach to true multi-image matching” *In Proc. CVPR96, pages 358-363, 1996.*

[52] http://cat.middlebury.edu/stereo/data.html, *last accessed 20.07.2007*

[53] http://research.microsoft.com/IVM/3DVideoDownload, *last accessed 05.01.2007*

[54] https://www.3dtv-research.org/3dav_CfP_FhG_HHI/, *last accessed 15..06.2007*

[55] X. Zabulis ans K. Daniilidis, “Multi-camera reconstruction based on surface normal estimation and best viewpoint selection”. *In Proceedings of the 2nd International Symposium on 3D Data Processing Visualization and Transmission, 2004.*

[56] C.Cigla, X.Zabulis and A.A. Alatan “Region-based dense depth extraction from multi-view video” *in International Conference on Image Processing, 2007.*

[57] Daniel DeMenthon. *“*Spatio-temporal segmentation of video by hierarchical mean shift analysis*”. In Porc. of Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, 2002.*

[58] M. Black, “Combining intensity and motion for incremental segmentation and tracking over long image sequences” *in European Conference on Computer Vision pp.485-493, 1992*

[59] W.B. Thompson, “Combining motion and contrast for segmentation”, *IEEE Transactions on PAMI, pp. 543–549, 1980.*

[60] D. Zhong and S-F. Chang, “Spatio-Temporal video search using the object based video representation”, *Proc. International Conference on Image Processing, Santa Barbara, CA, October 1997.*

[61] Y. Deng and B.S. Manjunath, “Unsupervised segmentation of color-texture regions in images and video”, *IEEE Trans. on PAMI, vol. 23, no. 8, pp. 800–810, 2001.*

[62] J. Y. A. Wang and E. A. Adelson, “Representing moving images with layers”, *IEEE Trans. on Image Processing, 3, pp. 625-638, Sept. 1994.*

[63] Tekalp M., Altunbasak Y., Eren E.,``Region-based affine motion segmentation using color information" *Graphical Models and Image Processing, vol. 60, no. 1, pp. 13-23, Jan. 1998*

[64] E.H. Adelson and J.R. Bergen, “Spatiotemporal energy models for the perception of motion”, *J. Opt. Soc. Am. A. 2(2), pp. 284–299, 1985.*

[65] R. Bolles, H. Baker, and D. Marimont, “Epipolar-Plane image analysis: an approach to determining structure from motion”, *Int. J. of Computer Vision, 1, pp. 7–55, 1987.*

[66] C. Fowlkes, S. Bellongie and J. Malik, "Efficient spatiotemporal grouping using the nystrom method", *CVPR 2001, Kaui, pp. I-231–238 , December 2001.*

[67] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts", *IJCV 98, Bombay, India, January 1998.*

[68] E. Francois and B. Chupeau, "Depth-based segmentation," *IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp. 237–239, Feb. 1997.*

[69] Ebroul Izquierdo M., "Disparity/Segmentation analysis: matching with an adaptive window and depth driven segmentation" *IEEE Transactions on Circuits and Systems For Video Technology, Vol. 9, No. 4, June 1999*

[70] W. Woo, N. Kim, and Y. Iwadate, "Object segmentation for Z-keying using stereo images" *in Proc. IEEE Intl. Conf. on ICSP, pp. 1249-1254, Aug. 2000.*

[71] Til Aach, Andre Kaup, "Disparity-based segmentation of stereoscopic foreground/background image sequences" *in IEEE Transactions on Communications, Vol: 42, No: 2, pg. 673-679 1994.*

[72] D. Markovic, M.Gelautz, "Video object segmentation using stereo derived depth maps" *In 27th Wokshop of the AAPR/ÖAGM, pages 197-204, Laxenburg, 2003.*

[73] P. An**,** C. Lii, Z. Zhang "Object segmentation using stereo images" *International Conference on Communications, Circuits and Systems, 2004. Volume 1, Issue , 27-29 June 2004 Page(s): 534 - 538 Vol.1*

[74] K. Conor and I. Reid, “A multiple view layered representation for dynamic novel view synthesis” *Proceedings of British Machine Vision Association, 2003*

[75] J. Schmitt et’al, “3D scene segmentation and object tracking in multiocular image sequences” *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007).*

[76] Horn, B.K.P. and Schunck, B.G., "Determining optical flow." *Artificial Intelligence, vol 17, pp 185-203, 1981.*

[77] B.D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision", *DARPA Image Understanding Workshop, 1981, pp121-130.*

[78] Michael Bleyer et.al, “Color-segmentation based computation of dense optical flow with application to video object segmentation” *Österreichische Artificial Intelligence Journal, 24(1), pp 11-15, 2005.*

[79] http://www.tanimoto.nuee.nagoya-u.ac.jp/, *last accessed 12.07.2007*

APPENDIX A

NORMALIZED CUTS FORMULATION

Assume a graph, V, with N nodes is partitioned into two disjoint sets A and B; and let x be the N-dimensional indicator, such that $x_i = 1$ if node i is in A and $x_i = -1$ otherwise. Let the total connection form node i to all other nodes is defined by $d(i)$ such that $d(i) = \sum_j w_{ij}$ where $w(i,j)$ indicate the link weight between nodes i and j. According to the definitions of x and d, the normalized cut value can be written as:

$$Ncut(A, B) = \frac{Cut(A,B)}{TotalW(A,V)} + \frac{Cut(A,B)}{TotalW(B,V)}$$

$$= \frac{\sum_{(x_i>0, x_j<0)} -w_{ij}.x_i.x_j}{\sum_{x_i>0} d(i)} + \frac{\sum_{x_i<0, x_j>0} -w_{ij}.x_i.x_j}{\sum_{x_i<0} d(i)}$$

Let the NxN diagonal matrix whose entries are $d(i)$'s be D and W be an NxN symetric matrix of $w(i,j)$'s such that $W(i,j) = w_{ij}$, then, define k as:

$$k = \frac{\sum_{x_i>0} d(i)}{\sum_i d(i)}$$

and $\bar{1}$ be an *N*x*1* vector with all ones. For $x_i>0$ and $x_j<0$, the indicator vectors can be defined as $\frac{\bar{1}+x}{2}$ and $\frac{\bar{1}-x}{2}$, hence *Ncut*(*A,B*) can be written as:

$$= \frac{1}{4}\left(\frac{(\bar{1}+x)^T(D-W)(\bar{1}+x)}{k(\bar{1})^T D\bar{1}} + \frac{(\bar{1}-x)^T(D-W)(\bar{1}-x)}{(\bar{1}-k)1^T D\bar{1}}\right)$$

$$= \frac{1}{4}\left(\frac{(x^T(D-W)x + (\bar{1})^T(D-W)\bar{1})\bar{1})}{k(\bar{1}-k)(\bar{1})^T D\bar{1}} + \frac{2(1-2k)(\bar{1})^T(D-W)x}{k(1-k)(\bar{1})^T D\bar{1}}\right)$$

The scale does not change the optimum partitioning; hence, it can be discarded. Let the following auxiliary variables be equal to,

$$\alpha(x) = x^T(D-W)x$$

$$\beta(x) = (\bar{1})^T(D-W)x$$

$$\gamma = (\bar{1})^T(D-W)1 = 0$$

and

$$M = (\bar{1})^T D1$$

then, the above equation can be expanded as follows:

$$= \frac{(\alpha(x)+\gamma) + 2(1-2k)\beta(x)}{k(1-k)M}$$

$$= \frac{(\alpha(x)+\gamma) + 2(1-2k)\beta(x)}{k(1-k)M} - \frac{2(\alpha(x)+\gamma)}{M} + \frac{2\alpha(x)}{M} + \frac{2\gamma}{M}$$

The last term can be dropped since it is equal to zero,

$$= \frac{(1-2k+2k^2)(\alpha(x)+\gamma)+2(1-2k)\beta(x)}{k(1-k)M} + \frac{2\alpha(x)}{M}$$

$$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(\alpha(x)+\gamma)+\frac{2(1-2k)}{(1-k)^2}\beta(k)}{\frac{k}{1-k}M} + \frac{2\alpha(x)}{M}$$

set $b = \frac{k}{1-k}$, then

$$= \frac{(1+b^2)(\alpha(x)+\gamma)+2(1-b^2)\beta(x)}{bM} + \frac{2b\alpha(x)}{bM}$$

$$= \frac{(1+b^2)(\alpha(x)+\gamma)}{bM} + \frac{2(1-b^2)\beta(x)}{bM} + \frac{2b\alpha(x)}{bM} - \frac{2b\gamma}{bM}$$

$$= \frac{(1+b^2)(x^T(D-W)x+\bar{(1)}^T(D-W)\bar{1})}{b\bar{(1)}^T D\bar{1}} + \frac{2(1-b^2)\bar{(1)}^T(D-W)x}{b\bar{(1)}^T D\bar{1}}$$

$$+ \frac{2bx^T(D-W)x}{b\bar{(1)}^T D\bar{1}} - \frac{2b\bar{(1)}^T(D-W)\bar{1}}{b\bar{(1)}^T D\bar{1}}$$

$$= \frac{(1+x)^T(D-w)(1+x)}{b\bar{(1)}^T D\bar{1}} + \frac{b^2(1-x)^T(D-W)(1-x)}{b\bar{(1)}^T D\bar{1}}$$

$$- \frac{2b(1-x)^T(D-W)(1+x)}{b\bar{(1)}^T D\bar{1}}$$

$$= \frac{\left[(1+x)-b(1-x)\right]^T(D-W)\left[(1+x)-b(1-x)\right]}{b\bar{(1)}^T D\bar{1}}$$

Set $y = (1+x)-b(1-x)$,

Thus, the normalized cut value can be formulated as:

$Ncut(A,B) = \frac{y^T(D-W)y}{y^T Dy}$ with the condition y(i)$\in${1,-b} and $y^T D\bar{1}=0$.

In addition, *b* satisfies the following property:

$$b1^T D1 = b(\sum_{x_i<0} d(i) + \sum_{x_i>0} d(i)) = b(\sum_{x_i<0} d(i) + b\sum_{x_i<0} d(i))$$

$$= b\sum_{x_i<0} d(i) + b^2 \sum_{x_i<0} d(i)$$

$$= \sum_{x_i>0} d(i) + b^2 \sum_{x_i<0} d(i)$$

$$= y^T Dy$$