Segmentation based depth extraction for stereo image and video sequence

by

Yu Zhang

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

Ottawa, Ontario, Canada

September 2012

Abstract

3D representation nowadays has attracted much more public attention than ever before. One of the most important techniques in this field is depth extraction.

In this thesis, we first introduce a well-known stereo matching method using color segmentation and belief propagation, and make an implementation of this framework. The color-segmentation based stereo matching method performs well recently, since this method can keep the object boundaries accurate, which is very important to depth map. Based on the implemented framework of segmentation based stereo matching, we proposed a color segmentation based 2D-to-3D video conversion method using high quality motion information.

In our proposed scheme, the original depth map is generated from motion parallax by optical flow calculation. After that we employ color segmentation and plane estimation to optimize the original depth map to get an improved depth map with sharp object boundaries. We also make some adjustments for optical flow calculation to improve its efficiency and accuracy. By using the motion vectors extracted from compressed video as initial values for optical flow calculation, the calculated motion vectors are more accurate within a shorter time compared with the same process without initial values.

The experimental results shows that our proposed method indeed gives much more accurate depth maps with high quality edge information. Optical flow with initial values provides good original depth map, and color segmentation with plane estimation further improves the depth map by sharpening its boundaries.

Contents

List of Tables

List of Figures

List of Acronyms

SAD	Sum of Absolute Difference
MRF	Markov Random Field
CRF	Conditional Random Field
GC	Graph Cut
BP	Belief Propagation
SD	Squared intensity Difference
NCC	Normalized Cross-Correlation
SSD	Sum of Square Difference
DIBR	Depth Image Based Rendering
PSNR	Peak Signal to Noise Ratio
ED	End Point error analysis
MPEG	Moving Pictures Experts Group
MV	Motion Vector
IMO	Independent Moving Object
CUDA	Compute Unified Device Architecture

This thesis is dedicated to my family.

Acknowledgement

This thesis was completed with the great help of my supervisor, my colleagues, my friends and my family.

I would like to first give me deepest gratitude to Professor Jiying Zhao, who not only guided me as my supervisor but also encouraged and challenged me throughout my academic program.

I would also like to thank our lab's visiting professor, Dr. Xuezhi Xiang. My final project would not completed so much successfully without his guidance, patience and encouragement.

Thanks so much to all my lab colleagues: Wenyi Wang, Zhenyi Luo, Chengcheng Hao, and Han Wang. I spent the most rich and happy life these two years because of you.

Most especially to my family, they give me the strongest and warmest support. All the successes and happiness cannot be tasted deeply without you.

Chapter 1

Introduction

1.1 Stereo vision

Stereo vision, which is a compelling topic currently, has been researched for a long time. In 1838, Charles Wheatstone [1] firstly provided the concept stereopsis, and invented stereoscope device to display his stereo pictures; after that, stereoscopy became popular in public with the invention of the prism stereoscope by David Brewster in Victoria time; until 1960s, researchers started their explore on stereopsis to find its limitations and its relationship to singleness of vision.

In stereo vision system, the different perspectives of two eyes lead to slight relative displacement of the same objects (disparities) for each eye, and this can introduce a depth sense to the brain (Fig. 1.1) . The human visual system transforms the slight sense displacement (pixel horizontal shifts) between the left-eye and right-eye into distance information such that objects are perceived at different depths and outside of the 2D display plane. In this case, the smaller the displacement, the farther we feel the object to us, and verse visa.

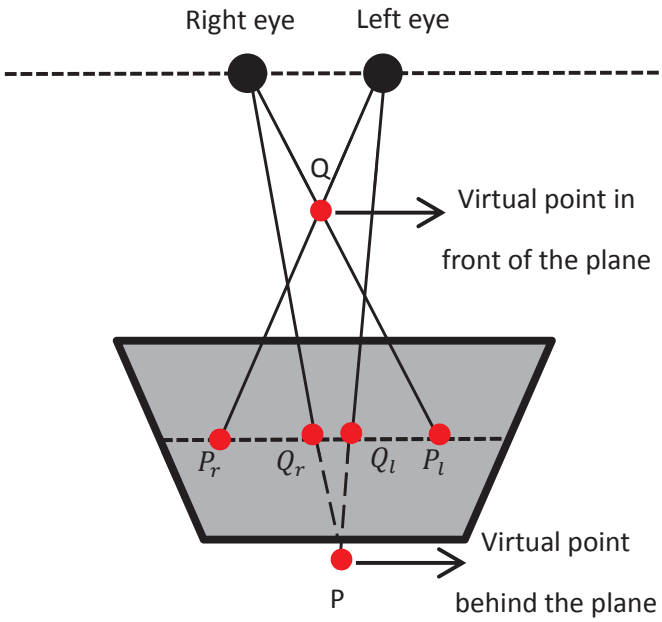


Figure 1.1: Depth representation from two views.

In current research, there are much more interests in inferring the disparity or depth map from the image pairs other than using image pairs directly to simulate the sense of 3D. The disparity map is an image that indicates the difference between the horizontal coordinate of two corresponding points from the image pairs, while depth map is an image that contains information relating to the distance between the surfaces of scene objects and a viewpoint.

In computer vision, disparity is often treated as synonymous with inverse depth (shown in Fig.1.2). The mathematic relation between disparity and depth are as follows:

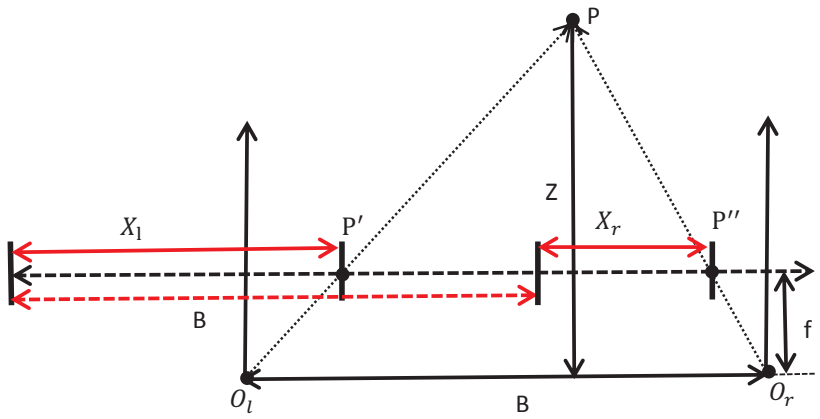


Figure 1.2: Depth via triangulation.

In Fig.1.2, B represents the baseline, which is the distance between focal points of two cameras. f is the focus length of the camera. X_l, X_r are the x coordinates of the corresponding points in the stereo images, which can be got from stereo matching algorithms. Z is the depth of the point in 3D space. From the similar triangular of $PP^{'}P^{''}$ and PO_lO_r, we can get:

$$\frac{b}{Z} = \frac{b - X_l + X_r}{Z - f}, \tag{1.1}$$

where $X_l - X_r$ is the disparity d. Then we can get the formula as follows:

$$Z = \frac{b \cdot f}{X_l - X_r} = \frac{b \cdot f}{d}. \tag{1.2}$$

The disparity information can be captured from the cameras or other devices directly, or using the image processing technology. To acquire the disparity information from devices, the optical sensors such as laser sensor, infrared ray sensor, or light pattern sensor are used. To acquire the disparity information via image processing technology, the

methods based on binocular disparity extraction and monocular disparity extraction are employed.

Monocular disparity extraction includes the techniques using motion parallax, focus/ defocus, linear perspective and atmospheric scattering, etc. Binocular disparity extraction includes stereo matching and convergence. (Fig. 1.3).

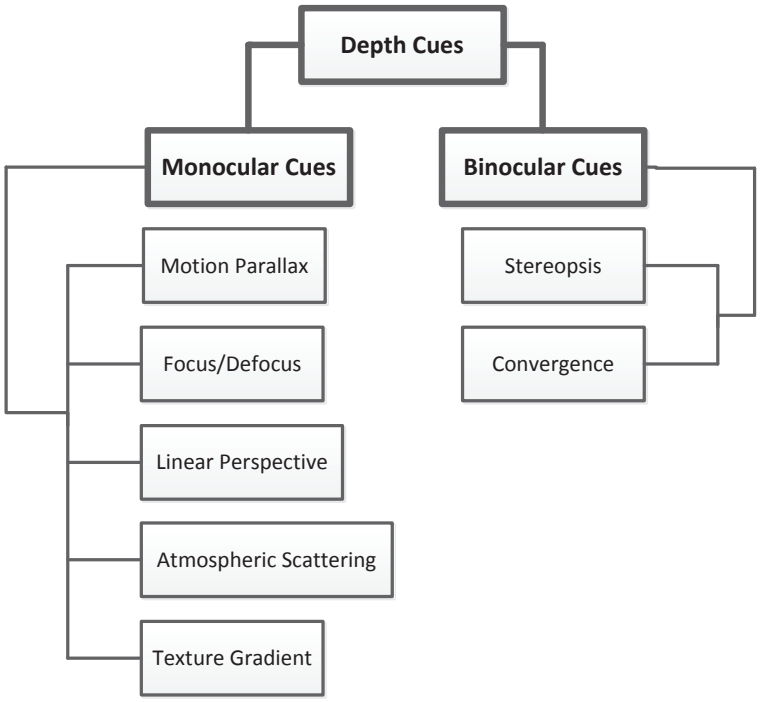


Figure 1.3: Classification of depth cues.

1.2 Binocular stereo matching method

Binocular stereo matching is one of the most classical research area in computer vision. Over the last 30 years, a broad range of algorithms for stereo correspondence have been developed. The goal of stereo matching is to determine the disparities that are indicating the difference in locating corresponding pixels.

When we solve the stereo matching problem, typically, we assume that corresponding pixels have the same intensity. However, in practical, image pairs always suffer the problems such as image noise, different illumination conditions between left and right images, specular reflection, and object transparent. What's more, according to the principle of stereo matching, a certain amount of intensity variation is required, i.e. texture, so that a pixel could be uniquely matched in a view. In order to put the stereo matching problem into an analytical level, computational theory and assumptions should be made. There are general five assumptions to make works go well. They are Lambertian surface assumption, Epipolar constraint, Continuity constraint, smoothness constraint, and Maximum disparity constraint.

1. Lambertian surface: The apparent brightness of such surface does not vary with viewpoint;

2. Epipolar constraint: To find the matching points between the image pairs, we only search along the epipolar line, which means the corresponding points have the same y coordinate;

3. Continuity constraint: Disparity tends to vary slowly and smoothness across a surface, so when we get two points on the left image which are located close to each other, then their corresponding point pairs on the right image should be also close to each other;

4. Smoothness constraint: Spatially adjacent pixel has the same (or similar) disparity;

5. Maximum disparity constraint: Every image has a probable maximum disparity, and this disparity value could be computed by getting the depth and geometry of

a stereo system. Mostly, we set the maximum disparity values as searching range when we start matching.

However, there are still some challenges in this area:

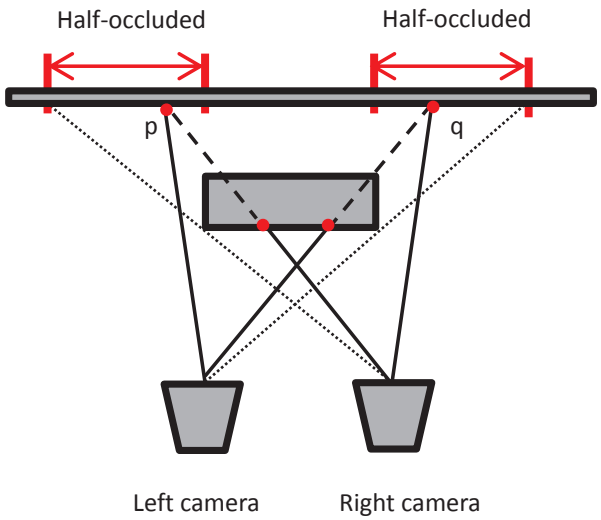


Figure 1.4: The occlusion problem.

1. Occlusion problem is a typical problem in stereo matching other than other computer vision problems. It is difficult to find the matching point which is hidden or does not exist, while ignoring occlusion problems may lead to disparity artifacts near object borders, as shown in Fig.1.4.

2. Another problem lies in the assumption of Lambertian surface. In real world, the assumption that a 3D point in space has the same appearance under projection from different geometries is not always true, even if image conditions remain ideal [2]. Some of these reasons are: A change of viewing angle will cause a shift in perceived (specular) reflection and color of the surface if the illumination source

is not at infinity or the surface does not exhibit Lambertian reflectance; focus and defocus may occur in different planes at different viewing angles if depth of field (DOF) is not unlimited; a change of viewing angle may cause geometric image distortion or the effect of perspective foreshortening if the imaging plane is not at infinity. Large depth variations of one point relative to its surrounding points may violate the ordering constraint or produce occlusions; a change of viewing angle or temporal change may also change geometry and reflectance of the surfaces if the images are not obtained simultaneously, but, instead, sequentially.

3. Problems also exist in region with constant color. The matching may fail in those areas. The lack of texture information makes it impossible to uniquely identify and match the pixels within the reference image.

For these reasons we can see that it is still a challenge and difficulty to get an accurate depth map from image pairs in practical situation. Hence, how to solve these problems and produce an accurate depth map is always an important work in the field of stereo matching.

1.3 2D to 3D video conversion

The vivid visual sense of 3DTV makes 3DTV be adopted successfully by the general public. However, the lack of the 3D contents is still a problem due to the complexity of shooting device. In this case, converting the existing 2D materials into 3D format is a better and more convenient way to obtain 3D video content.

2D to 3D video conversion basically is an important application in computer stereo vision. There are mainly two differences between binocular stereo matching and 2D-to-3D video conversion. Firstly, 2D-to-3D video conversion extracts the depth from the

consecutive image sequences by using the motion information and geometric information of frames. While stereo matching gets the depth from image pairs, mainly uses local block matching and global energy minimisation method. Secondly, 2D-to-3D video conversion, unlike stereo matching, dose not need to make the rectification to the images before processing, because it is unreasonable and meaningless to make any changes to the processed video for depth generation. In this way, 2D-to-3D video conversion is an extension of stereo matching on both theories and techniques.

According to the survey by Zhang [3], 2D to 3D video conversion methods can be classified into three groups: manual scheme, human-assisted scheme and automatic scheme.

The manual scheme normally shifts the pixels in horizontal direction with the given depth values which are manually set, and produces a new image. The human-assisted scheme is to convert 2D content into 3D with the help of manual operators. The automatic scheme extracts the depth information from the 2D video or image sequences directly. Although the manually scheme can provide the depth with the highest quality, this process is time consuming and expensive with the engagement of human. The human-assisted scheme reduces the humans work, however, it still needs the human to complete the entire conversion work. The automatic scheme totally avoids human engagement during the conversion process, however, it produces the lowest quality of depth. Even in this case, the automatic scheme attracts more interests in public due to its automation and efficiency.

In an automatic 2D to 3D video conversion scheme, the framework basically consists of two parts (Fig.1.5): the extraction of depth information from image sequence and the synthesis of stereo images according to the generated depth information and view conditions. The first step is to get the depth information according to the geometric

information or motion parallax of image sequence. The second part is to convert the depth information to a suitable representation for usage.

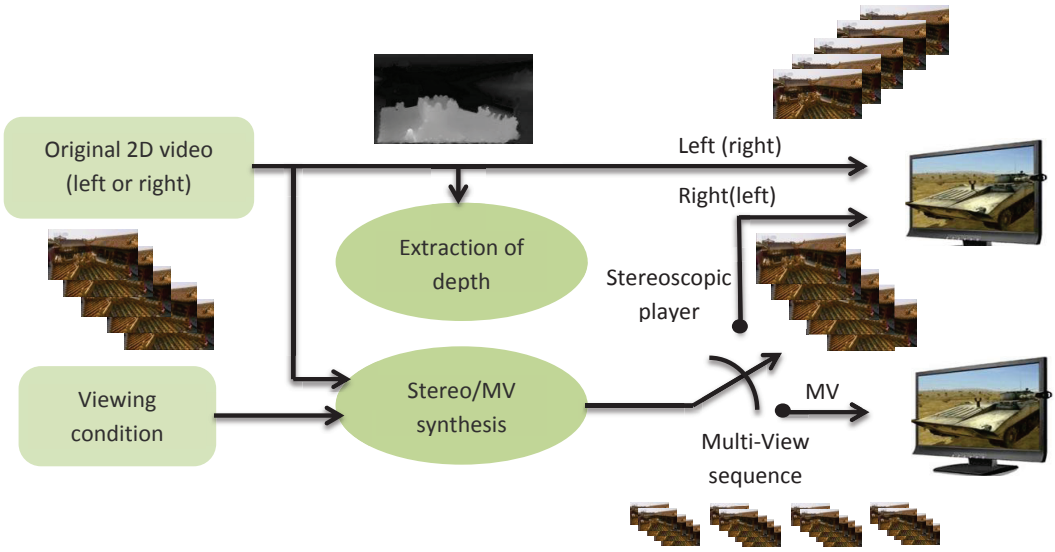


Figure 1.5: Automatic 2D-to-3D conversion system.

Recent automatic 2D-to-3D conversion schemes can be categorized into two groups: pictorial cues based scheme and motion based scheme. The pictorial cues based schemes operate on a single still image, and use the geometric and color information from image to construct the depth of a scene. Those pictorial cues include linear perspective, patterns texture and occlusion, etc. Motion based scheme use the motion parallax provided by video sequence to generate depth. In motion based scheme, methods such as extracting motion vectors from compressed video, getting motion parallax from feature tracking of image sequence, and getting motion information from optical algorithms are mainly used.

1.4 Contributions of the thesis

In this thesis, we first introduce a well-known color-segmentation based stereo matching method proposed by Klaus [4], and then make an implementation of this scheme according to our own understanding. Since there is no implementation description introduced in the original paper, our work makes contributions by understanding the principle and techniques in each step. We give a clear explanation for how to implement each function, algorithm and technique of the scheme, how to set and adjust parameters, how truncation values work, and why we deal with some details in a specific way. In addition, the intermediate results shown in our implementation made this method more understandable to the researchers in this field.

Second, we proposed a 2D-to-3D video conversion scheme based on our implemented framework. In our scheme, an automatic method based on color segmentation and high quality motion information is proposed. We use optical flow algorithm to get the motion vectors from the consecutive frames. In comparison to setting the motion vector directly from compressed video sequence, optical flow algorithm has its own pros and cons. Optical flow calculation can provide much more accurate motion information than extracting motion vectors from compressed videos. However, it is a little time consuming and complexity. In this way, in order to improve both the efficiency and accuracy of the optical flow calculation, we take motion vectors from compressed video as the initial input for optical flow calculation. In addition, color segmentation and plane fitting are employed to get sharp boundaries of objects.

The performances of the proposed scheme are impressive. From the results we can see, the proposed scheme successfully produces high quality depth with sharp boundaries.

Publications generated from the research:

[1] Yu Zhang, Wenyi Wang, Xuezhi Xiang, and Jiying Zhao, "Segment-Based Stereo Matching using Belief Propagation : an implementation," IEEE Conference on Electrical and Computer Engineering, April 29 - May 2, 2012, Montreal, Canada.

[2] Yu Zhang, Xuezhi Xiang, and Jiying Zhao, "2D to 3D Video Conversion Based on Color Segmentation and Hight Quality Motion Information," ACM Multimedia 2012, October 29 - November 2, 2012, Nara, Japan.

1.5 Thesis structure

In Chapter 2, an introduction of current stereo matching methods is presented, followed with a comprehensive review of 2D to 3D video conversion schemes; Chapter 3 describes the structure and corresponding principles of Klaus' scheme, and presents the implementation process step by step according to our own understanding; Chapter 4 gives the structure and principles of our proposed 2D to 3D video conversion scheme; Chapter 5 illustrates the experimental results and makes the analysis of our proposed scheme; Chapter 6 concludes the thesis and gives the suggestions for our future work.

Chapter 2

Literature review

2.1 Binocular stereo matching schemes

2.1.1 Local and global methods

According to a comprehensive survey made by D. Scharstein and R. Szeliski [5], most stereo matching algorithms generally perform the following four steps: matching cost computation, cost (support) aggregation, disparity computation/ optimization, disparity refinement. And based on that, methods for stereo matching could be roughly classified into two categories: local and global methods.

For local methods, the disparity computation at a given point depends only on intensity values within a finite window. The matching cost is aggregated over the window, and the disparity value with the minimal cost is selected as the corresponding disparity of the pixel. Common approaches to match the local windows are to use some matching criterions such as squared intensity difference (SD), absolute intensity difference (AD), normalized cross-correlation (NCC), and sum-of-square-difference (SSD). These window-based methods produce results with pixel-level accuracy. The limitations of

local methods are obvious: firstly, the local minimums may not guarantee the global minimum. Also, due to the lack of content information within a searching window, the improvement of local results must rely on wider searching space. However, as the searching space expanding, the geometric effect becomes more and more noticeable which can wrongly affect the final result. Even though local method have certain disadvantages, it is employed in many systems with real time requirement for its low computational complexity [6] [7].

For global methods, they perform almost all the computations except the aggregation step. Many global methods are formulated in an energy-minimization framework. The aim for global method is to find a disparity function which minimizes a global energy over the image. By reviewing the recent methods, global method has become the dominant technique to solve the stereo matching problem. Since the corresponding problem can be modeled as a Markov Random Field (MRF) or Conditional Random Field (CRF) optimization problem, lots of global methods based on that were proposed. Methods like Graph Cut [8] [9], Belief Propagation [10], and QPBO [11] [12] use energy functions to describe the relation between views, and assume that minimizing such energy function the most will produce the best correspondence. And most state-of-the-art algorithms in the Middlebury benchmark [13] fall into this category. However, the expensive computational cost, which introduced by the slow-converging optimization process, is the major limitation of this kind of matching algorithms.

2.1.2 Segmentation based global methods

Segmentation technique has become more and more popular in recent years in binocular stereo matching, since it can yield accurate boundaries of objects which is very important to depth map. Based on the well-known Middlebury benchmark, most of

the top-ranked methods used image segmentation in different way in their proposed schemes.

Klaus [4] proposed a color segmentation based method using belief propagation and a self-adapting dissimilarity measure to produce depth map. This method achieved an outstanding result compared to most of the other methods ranked in Middlebury benchmark. It is based on the assumptions that the scene structure can be approximated by a set of non-overlapping planes in the disparity space and that each plane is coincident with at least one homogeneous color segment in the reference image. Based on the initial disparity map produced by improved block-matching method and label map produced by color segmentation, the disparity map is updated by using plane estimation. This method also provides a basic framework for our research.

Wang [14] continued with the color-segmented method and made some changes at disparity optimization step. In their paper, an inter-regional cooperative optimization procedure was employed to minimize the matching costs of all regions. They defined an energy function including data term, smoothness term and occlusion term and used cooperative and competitive relations between adjacent regions to optimize the total image energy function. It is still a top-three method in Middlebury benchmark. What's more, this scheme does very well in restraining and correcting mismatched errors.

Those two methods above represent typical and best color-segmented methods up to now. However, the key drawback of all these approaches is that the segmentation assumption is a hard constraint, i.e. two pixels which share the same segment must be assigned to the same plane. To make the optimization, Bleyer [15] proposed a new stereo matching model based on a simple assumption that a scene is composed of a few and smooth surface. They proposed a soft constraint assuming that self-similar pixels would lie on the same surface. The experimental results in Bleyer's paper

showed that his scheme can correctly capture the disparity discontinuities due to the soft segmentation term and occlusion handling.

After one year, Bleyer [16] proposed the idea of combining segmentation and stereo matching to the next level: the object-level. They extended ideas of object-level segmentation in 2D images to 3D scenes. In their scheme, each object is characterized by color model, 3D plane which approximates the object's disparity distribution and novel 3D connectivity property. The novel contribution of their scheme was that small untextured region with homogeneous color could be assigned the disparities, which means their scheme can handle the occlusion problems very impressively. Fig.2.1 shows their remarkable work on occlusion case.

2.1.3 GPU based stereo matching methods

Although segmentation based techniques can produce accurate disparity maps, they still need a considerable running time. Considering matching accuracy and the processing efficiency, the real-time stereo matching implemented on GPU become the new trend in nowadays.

Zhang [17] presented a CUDA based local matching method using bitwise fast voting in their paper in high efficiency. They made the cost aggregation over a shape adaptive support window and used the reliable initial estimates to make the refinement. They also proposed a sample-and-restore scheme to speed up this scheme with the minimum accuracy degradation. Their scheme was among the fastest stereo matching methods on GPUs.

Mei [18] proposed a GPU-based stereo matching system with good performance in both accuracy and speed. In their scheme, an AD-census measure [19] they proposed previously was employed to acquire the initial matching cost; and the aggregation was

(a) Input image.

(b) Extracted objects.

(c) Generated depth map.

Figure 2.1: Results on Teddy image. Their algorithm propagates reasonable depth information to completely occluded regions using color information. (figures are cited from [16].)

processed in the adaptive window; then multi-step refinements were used to produce the final results. Each step of the proposed system was designed parallel so that the computations can be accelerated with CUDA implementations. Currently it is the top performer in the Middlebury benchmark, and the results are achieved on GPU within 0.1 seconds.

Kowalczuk proposed [20] a real time stereo matching method this year using two−pass approximation of adaptive support-weight aggregation, and a low-complexity iterative disparity refinement technique. Also this method has been implemented on massively parallel high-performance graphics hardware using the CUDA computing engine. From their statement, their results are the most accurate method among all of the real-time stereo matching methods ranked on the Middlebury benchmark.

2.2 Automatic 2D to 3D video conversion schemes

The recent state-of-the-art automatic 2D-to-3D conversion schemes could be categorized into two groups: motion based schemes and pictorial based schemes. For motion based schemes, fixed cameras located at different viewing spots, or single camera with moving objects are used to extract the depth information from motion parallax; the second group operates on a single still image by taking advantages of those pictorial cues such as linear perspective, patterns texture, defocus, occlusion, etc.

2.2.1 Motion based methods

Motion parallax refers to the relative motion between the viewing camera and the observed objects. For a moving observer, nearer object moves faster than the farther ones do. In this case, motion parallax provides an important cue to depth perception.

Normally, in motion based schemes, the motion parallax will be firstly determined from the video sequence, and then the motion parallax will be mapped into the depth information.

In principle, not all the video sequences could provide correct motion parallax to the depth, only those who are captured by a moving camera with still scene can provide correct motion parallax. When the camera is fixed, even though the scene appears some Independently Motion Objects (IMOs), the motion information from those IMOs will also violate the principle of depth from motion. Since IMOs have relative motions which are different from those of the background with respect to the camera.

Motion based 2D to 3D video conversion schemes can be generally classified into direct and indirect models in Zhang's survey [3]. Direct methods extract the motion information directly from the images, while indirect methods estimate the motion by tracking separate feature in the image sequence.

For direct method, extracting motion information from the compressed video sequence is a popular way. Since the motion vectors (MVs) could be extracted directly from the compressed video, the computational cost could be reduced. Ideses [21] presented a 2D to 3D video conversion method by taking advantage of the compressed video. In their scheme, the objects' motions are assumed to be proportional to their distances from the camera, and the motion information is extracted by using a fixed block size from the compressed video. Po [22] also took a similar way to get the motion information, and acquired sharper object boundaries of depth map with color segmentation. Pourazad [23] only took the horizontal component of motion vectors extracted from H.264/MVC, and further improved the accuracy of motion vectors via a nonlinear model. After one year, he provided a more effective scheme via matching blocks with various sizes to get the motion information of the H.264 encoded video and their

proposal improved the accuracy of the motion vector from MPEG4 [24]. (Similar work can still be found in Min [25].)

Besides, optical flow based methods are another direct way to make the motion estimation. Lee [26] presented a method taking Horn and Schunck [27] optical flow algorithm to make the depth estimation between two scenes; Seon-Yeong [28] employed Lucas and Kanade optical flow algorithm [29] and Mean Shift algorithm [30] to realize the 3D virtual target and surround environment models using 3D location information. In their scheme, motion information produced by optical flow algorithm could be used to make the depth estimation, and the mean shift algorithm is used to separate the target from its surroundings;

For indirect methods, motion parallax is estimated through image features. Zhang [31] created depth perception by using an efficient and robust face tracking algorithm to estimate the location and scale of faces, combined with foreground/background segmentation and matting algorithm, which is robust to moving background, lighting variations, and moving camera. Kim [32] proposed a MRF-based initial contour tracking and graph-cut-based contour refinement method. In their scheme, KLT (Kanade-Lucas-Tomashi) feature points are first extracted from previous frame, and then their corresponding points are found in current frame. In this way, depth map is formed according to the tracking.

2.2.2 Pictorial cues based methods

Pictorial depth cues originally have been applied in visual arts for centuries. Those cues such as the relative height, perspective and atmospheric scattering will provide the structure information of a scene to enhance the perception of depth. In contrast to the motion parallax, pictorial cues extract the relative depth other than absolute depth

of the objects because of the lack of knowledge for position and optical characteristics of camera.

One cue of pictorial is depth from focus and defocus. According to the human visual system (HSV), human eyes make accommodations when focusing on different planes in depth. This mechanism could be employed to generate the depth information from the images which contain both focused plane and objects out of the focused plane. Two approaches are employed normally: one is extracting the blur information from single image, and mapping the blur information to the depth plane. It is the first mechanism to be used to extract the depth from images [33]. Hariharan [34] provided a method recently based on focusing by using tensor voting to exploiting scene geometry and focus information to generate a reliable depth map comparing. Another is using several images with different focus to extract blur variation, and mapping the variation into depth. Yang [35] proposed virtual focus and used multiple images in video sequence captured by out-of-focus camera to make the blur estimation, and then generated the in-focus image sequences from those blur information.

Linear perspective is another important cue which refers to that the parallel lines will reach their convergence at infinite distance, and the farther the objects are, the more the lines converge. By detecting the parallel lines and the vanishing points of images, the suitable depth values can be assigned based on the positions of the lines and points. Yu [36] used linear perspective to reconstruct the static background of a scene; Ding [37] used multi-perspective images acquired from General Linear Cameras, and a novel Graph-Cut-Based algorithm to make a reconstructed for 3D scene.

Atmospheric scattering, also known as haze, refers to the light rays propagation will be affected in their power and directions by the particles in atmosphere. This will lead to different visual feelings: near objects look more distinct than the distant ones. Dark

channel prior, proposed by He [38] is a concept produced from this. It is drawn from a statistic analysis of outdoor haze-free images. The results of He are impressive, and the side product of their system is depth map. After that, lots of methods based on dark channel prior were proposed [39] [40] [41] [42] with satisfactory results.

Aside from the pictorial cues mentioned before, relative height, occlusion and texture gradient are also used to produce the depths. The relative height assumes that the bottom objects in image are more likely to be closer than the top ones [43]; depth-from-occlusion is based on the phenomenon that overlapping or partly observed objects are considered to be further [44]; texture gradient produces the depth based on the assumption that the gradient value of nearer object is larger than the farther ones [45].

2.2.3 Hybrid 2D to 3D video conversion models

Recent years, some hybrid models for depth generation from 2D video sequence were proposed and their performances were impressive. Lin [46] categorized a video into three groups based on the video content, and each group had a different conversion method. Basically, the conversion methods for those three groups are a combination cues analysis of motion parallax, atmospheric perspective, texture gradient, linear gradient and relative height. Battiatoa [47] also made a classification of the video sequence. In their scheme, geometric depth map and qualitative depth map were first constructed, and then final depth map was generated based on these two intermediate depth maps. Pourazad [48] proposed a scheme using Random Forest machine learning technique to make the conversion. Based on multiple cues such as motion parallex, haze, perspective and texture gradient, they tried to model a depth map of the recorded scene.

Chapter 3

The framework and implementation of segmentation based stereo matching

This chapter describes the principles and the implementation of Klaus' scheme. Since the proposed scheme is basically based on Klaus stereo matching scheme. This implementation provides us a good platform for 2D to 3D video conversion. The structure of Klaus' scheme is shown in Fig.3.1

Klaus' scheme consists of four steps to generate final result: color segmentation to reference image to get the labeled map, block matching using self-adapting method to get the original disparity map, plane fitting and belief propagation to make the refinement to get the final disparity map. And the test image pairs we used in the implementation are "Tsukuba" (282×382), "Venus" (383×434), "Cones" (375×450) and "Teddy" (375×450) from the Middlebury benchmark [13].

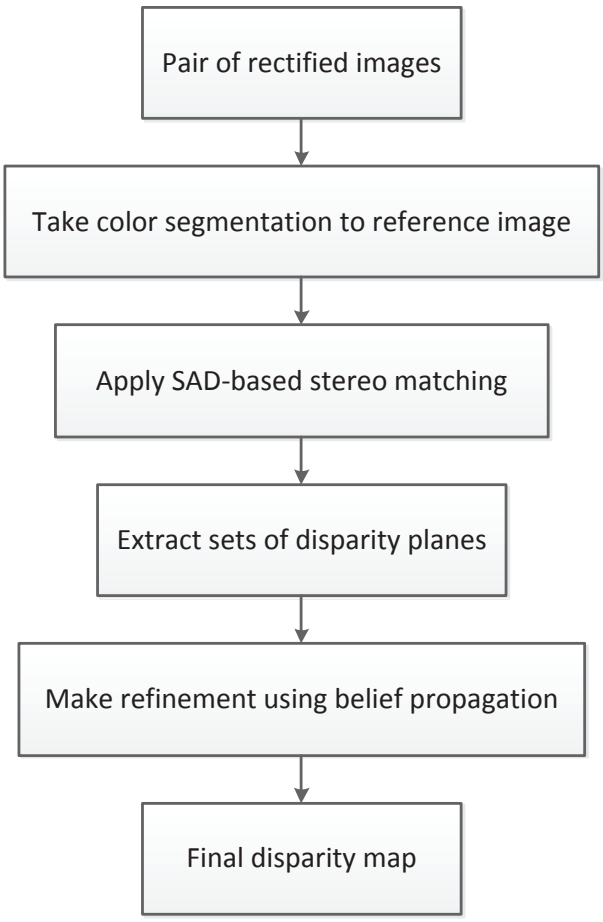


Figure 3.1: The structure of Klaus' scheme.

3.1 Color segmentation to extract homogenous regions

Color segmentation method has been used a lot in binocular stereo matching in current years due to its good performance in keeping the object boundaries. It is based upon the assumption that the content of image can be constructed by a group of non-overlapping planes in the depth space, each plane is supposed to contain at least one homogeneous color region, and the discontinuity only occurs on region boundaries. In this way, edge

information of image can be kept well. In Klaus scheme, mean shift algorithm is used to make the segmentation.

Mean Shift was introduced by Fukunaga and Hostetler [49] and has been extended to be applied in fields like Computer Vision. It treats feature space as empirical probability density function. The cluster presented in the feature space will correspond to the local maxima of the density function, thus we can classify clusters associated with the given local maxima using Mean Shift. Fig.3.2 shows how mean-shift algorithm makes the clustering.

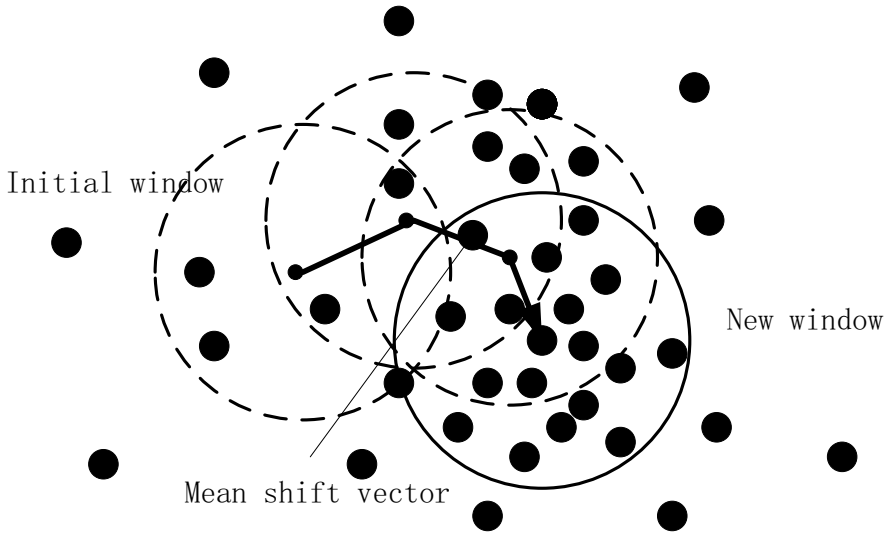


Figure 3.2: The principle of mean shift.

The mean shift algorithm works as follows: for each data point in the color space, it firstly calculates the mean value within a window, and then shifts the central of the window to the mean. By repeating this progress, the central of the window will reach its convergence and the points having high relation with it will be clustered together.

Mean shift is a non-parametric iterative algorithm and can be considered to be based

on Gradient Ascent on the density contour. The basic gradient ascent formula is:

$$x_1 = x_0 + \eta f^{'}(x_0), \tag{3.1}$$

Kernel density estimation is a non-parametric method to estimate the density function of random variables. Given a kernel K, bandwidth parameter h, Kernel density estimator for a given set of d-dimensional points is:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K(\frac{x - x_i}{h}), \tag{3.2}$$

Combining (3.1) and (3.2), we can get:

$$\vec{x} = \frac{\sum_{i=1}^{n} K^{'}(\frac{x-x_i}{h})\vec{x_i}}{\sum_{i=1}^{n} K^{'}(\frac{x-x_i}{h})}, \tag{3.3}$$

As we have mentioned, mean shift considers the data points in feature space as probability functions. Based on (3.3), assuming $g(x) = K^{'}(x)$, then we have:

$$m(x) = \frac{\sum_{i=1}^{n} g(\frac{x-x_i}{h})x_i}{\sum_{i=1}^{n} g(\frac{x-x_i}{h})} - x, \tag{3.4}$$

where $m(x)$ is the mean shift, and the clustering progress can be summarized as: for each data points xi, we first calculate the mean shift vector $m(x)$, and then move the density estimation window by $m(x)$. And then repeat this progress until convergence.

The mean shift algorithm is implemented using C++. The color segmentation results are shown in Fig.4.3.

There are three arguments $(sigmaS, sigmaR, andminRegion)$ used in our program. Among them, sigmaS and sigmaR present the spatial radius and range radius of the mean shift window respectively; minRegion indicates the minimum density a region may

(a) The original frame. (b) Color segmented frame.

Figure 3.3: Color segmentation to the reference frame.

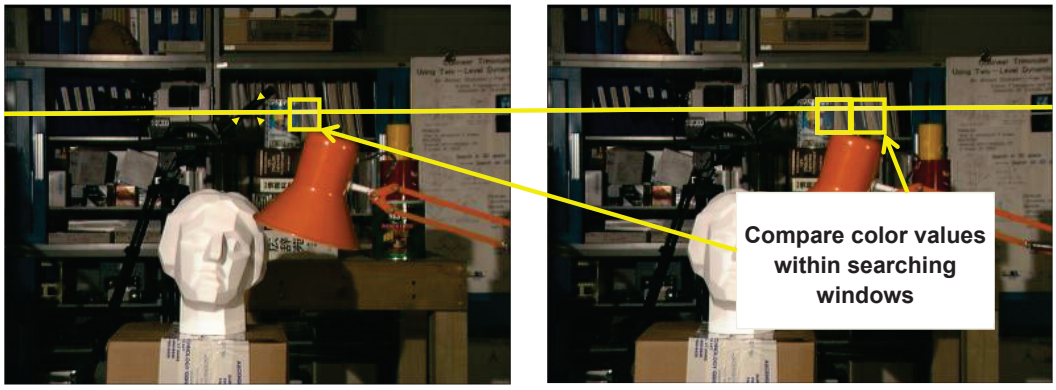
have in the resulting segmented image. According to our experiments, the parameters need to be changed according to the content of image. After repeating tests to these data sets, we found that the parameters given in Table 3.1 can achieve better results.

Table 3.1: The parameters for image sets

Parameter	*sigmaS*	*sigmaR*	*minRegion*
Tsukuba	4	4	20
Teddy	8	8	30
Cones	8	8	30
Venus	2	2	15

3.2 SAD-based stereo matching

Stereo matching based on the assumption of brightness consistency between left image and right image, which means the brightness of the same object will not vary much with the changes of view perspective. SAD-based block matching is local based stereo matching method. Fig.3.4 presents how it works:



(a) Left image (b) Right image

Figure 3.4: The principle of block based matching.

For each pixel in left image, we firstly place a small window surround it and match the whole window on the same horizontal scanline in the right image, and then we select the pixel with the most similar color in the right image as matching point. By repeating this progress to the whole image, we can get a raw disparity map.

The basic SAD-based block matching function is:

$$C_{SAD}(x,y,d) = \sum_{(i,j)\in N(x,y)} I_1(i,j) - I_2(i+d,j), \tag{3.5}$$

where (i,j) indicate the pixel's location, d represents the disparity value.

In Klaus' scheme, self-adapting dissimilarity measurement that combines sum of absolute intensity differences (SAD) and a gradient difference based measure is used. The gradient based measurement is defined as follows:

$$\begin{aligned} C_{GRAD}(x,y,d) = &\sum_{(i,j)\in N_x(x,y)} |\nabla_x I_1(i,j) - \nabla_x I_2(i+d,j)| + \\ &\sum_{(i,j)\in N_y(x,y)} |\nabla_y I_1(i,j) - \nabla_y I_2(i+d,j)|, \end{aligned} \tag{3.6}$$

where $N(x, y)$ is a $3*3$ surrounding window at position (x, y), $N_x(x, y)$ is a surrounding window without the rightmost column, $N_y(x, y)$ is a surrounding window without the lowest row, ∇_x is the forward gradient to the right and ∇_y is the forward gradient to the bottom. Color images are taken into account by summing up the dissimilarity measures for all channels.

The final cost function is the combination of SAD cost and gradient cost, which is defined as follows:

$$C(x, y, d) = (1 - \omega) * C_{SAD}(x, y, d) + \omega * C_{GRAD}, \tag{3.7}$$

where C_{SAD} is the intensity matching cost, C_{GRAD} is the gradient matching cost, (x, y) indicates the location of pixel, and ω is a weight value.

The weight value can be determined by experimental way when the final cost value reach the lowest, which mean the number of corresponding points reach the maximum. After that, cross-check will be done to improve the reliability of the matching points. The progress of cross-check is as follows: we first use the left image as reference image to get a disparity map, and then we use right image as reference image to get another disparity map. By comparing those two disparity maps, we will remove those unreliable points which do not obey the cross-checking function. The cross-checking function is:

$$D(x, y, r) = D(x + D(x, y, r), y, l), \tag{3.8}$$

where $D(x, y, r)$ represents the disparity map from right image, $D(x, y, l)$ represents the disparity map from left image.

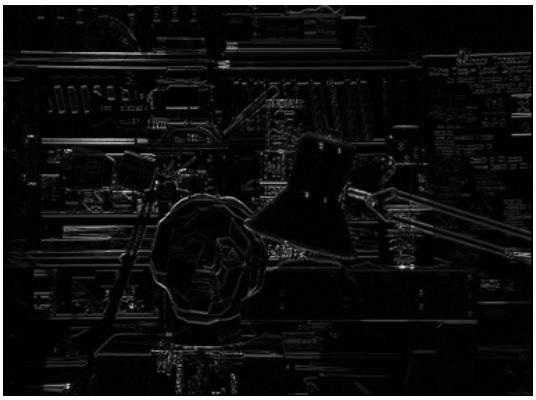
In our implementation, firstly, a matching cost threshold is set when calculating SAD within a window to choose the correct disparity. If all the matching costs within

the search range are higher than this threshold, the disparity of this pixel will be set to zero. In this case, those occlusion regions which do not have their matching points from corresponding image will not get wrong disparity values.

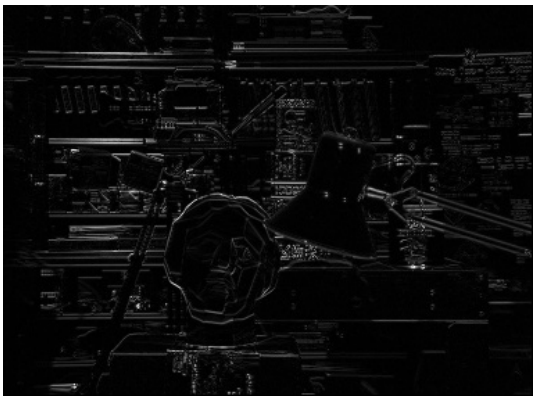
According to our experiments, the following parameters can produce the best results. The weight value, which controls the effect of the gradient components, is 0.9. The searching region for Tsukuba is set to 16 , and the searching region for the other three test sets is set to 45. The window size is set to 3. The implementation results of this step are show in Fig.3.5.

After cross-checking, the raw disparity map is generated. In our implementation, to enhance the disparity value's reliability and reduce the mismatched disparity, we further add a step to deal with the raw disparity map. For each segment, we employ statistical method to calculate the occurrence probability of disparity values in this segment and choose those with higher probability as reliable value. And then set the disparity of those points with low occurrence probability to zero, so that the outliers in one segment can be removed. According to our experiments, the threshold value to choose the disparities within a region is 0.2. The results of this step are shown in Fig.3.6.

(a) Left horizontal gradient map.

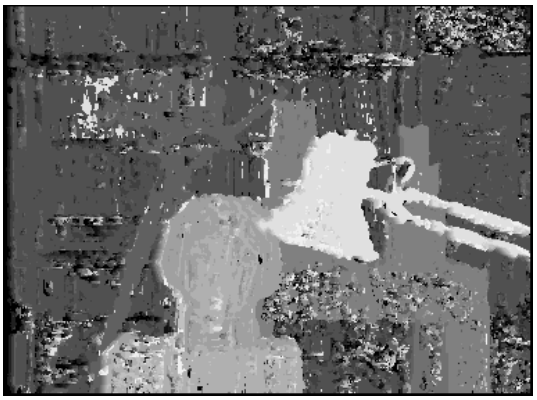


(b) Right horizontal gradient map.

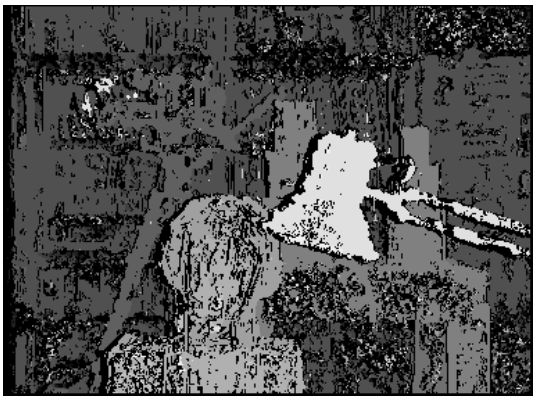


(c) Left vertical gradient map.

(d) Right vertical gradient map.

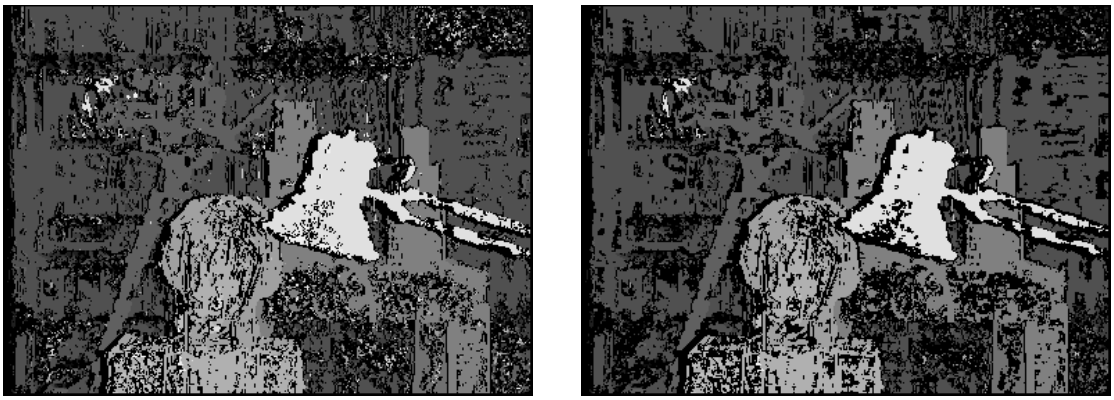


(e) Block matched disparity map.



(f) Cross-checked disparity map.

Figure 3.5: Gradient maps and raw disparity map for Tsukuba. (Window size =3, Searching range =16, Threshold =1.2, Weight value =0.9.)



(a) Before adjustment. (b) After adjustment.

Figure 3.6: Disparity adjustment.

3.3 Plane fitting within a homogenous region

In Klaus' scheme, the disparity planes are built based on the raw disparity map using plane fitting. Plane fitting is an algorithm to estimate a planar plane using three parameters. For each plane, the disparity values can be described as:

$$d = c_1 x + c_2 y + c_3, \tag{3.9}$$

where d presents plane's disparity, (x, y) indicates the pixel location. Parameters c_1 and c_2 present the gradient values of horizontal slant and vertical slant respectively.

The most popular way to determine the parameters of a plane is to solve a least square system. However, least square system is too sensitive to the outliers. In Klaus' scheme, they choose to use a linear or median solution to solve the outlier's problem.

In their scheme, the horizontal slope c_1 is the average difference between the adjacent pixels in horizontal direction. Similarly, the vertical slope c_2 is the average difference between the adjacent pixels in vertical direction. After the determination of c_1 and c_2, c_3 can be calculated by averaging the c_3 at each pixel:

$$C_3 = d - c_1 x - c_2 y, \tag{3.10}$$

$$c_3 = \frac{\sum_{c3 \in R} C_3}{N}, \tag{3.11}$$

where R is the current homogenous region, N is the number of the reliable points within this region.

In this way, the three parameters are determined, and a segment can be then estimated.

In our implementation, function REGRESS (Y, X) from Matlab is used to make linear plane fitting. The assumption of linear plane fitting is that the depth of object surface changes smoothly and linearly. In this case, we firstly transform the disparity value to depth value and then plane-fit the discrete reliable depth values. After plane fitting, the depth map should be transformed back to disparity map to generate the output.

The input parameters of this algorithm have been obtained from the previous steps. The color segmentation prepares the label map and the number of different segments in the color image; the SAD block matching provides the raw disparity map which will be refined in this step. Given the pixels with reliable raw disparity value, and the segment to which these pixels belong to, the output of plane fitting can be generated. Two truncation values are used to contribute to the high reliability in this part. Considering the reliable disparity values in a small segment may very low, we set the disparity values to zero in these regions. The first truncation value is used to make this judgment. The second one controls the accuracy of plane fitting. When the sum of the residual between the original disparity and the fitted disparity within one segment is higher than this truncation value, its disparity will be set to zero. The results are shown in Fig.3.7.

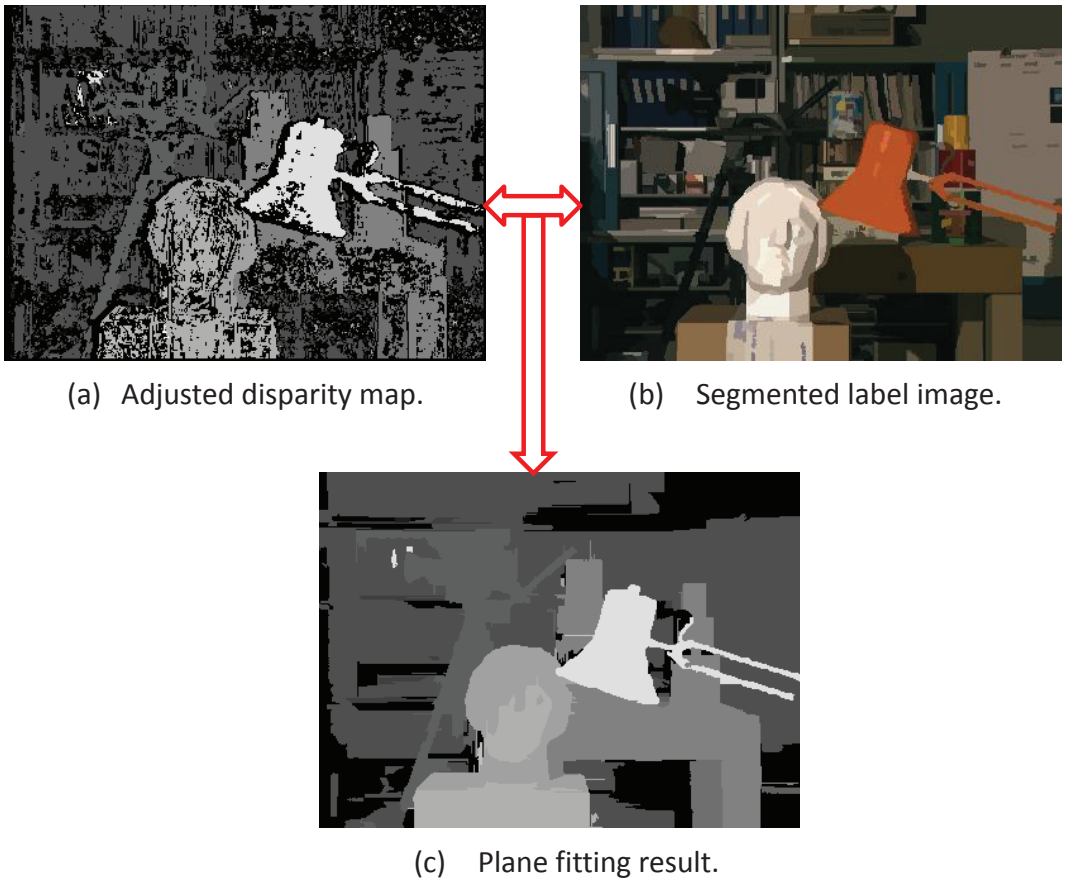


Figure 3.7: Plane fitting based on adjusted disparity map and segmented map.

3.4 Belief propagation to make refinement

Belief propagation is a widely used optimization algorithm for minimizing energy function. It is a message passing algorithm to solve inference problems based on Bayesian networks or Markov random fields. And nowadays, this technique has been developed and demonstrated on solving problems like stereo and image restoration.

With regard to the application in stereo matching, belief propagation is a progress shown is Fig. 3.8. The red circles represent pixel points and each pixel point has a belief about its optimal disparity assignment. By sending the messages to their

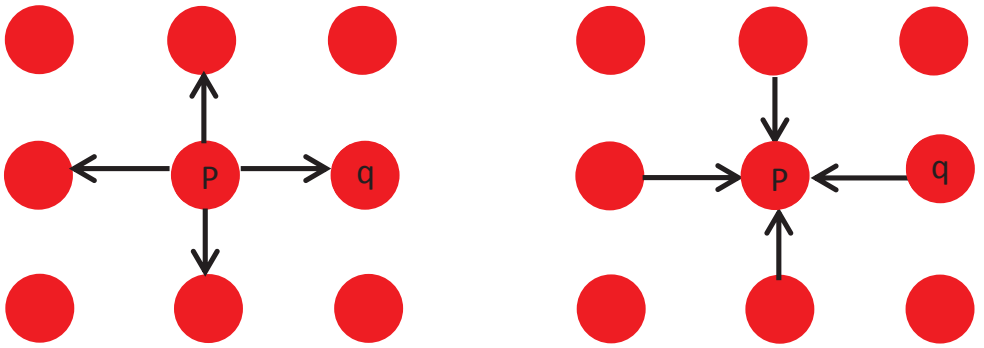


Figure 3.8: The principle of belief propagation.

neighborhoods and receiving messages from their neighborhood about what they think their neighborhoods' disparities are, the pixels will update the believes about their own disparities to reach a consistency with their neighborhoods. This is an iterative progress. After certain iterations, all the pixels will be assigned to reasonable and global agreements' disparities to cost minimum energy.

Mathematically, belief propagation is a max-product algorithm. It works by sending messages around the four-connected image grid and all the messages will be passed in parallel. Let $m^t_{p\to q}(d)$ be the amount of belief to which pixel p thinks pixel q is assigned to disparity d at iteration t and $m^0_{p\to q}(d)$ is initialized to zero. Then the updated messages at each iteration is defined as follows:

$$m^t_{p\to q}(d) = \min_{d'} s(d, d') + m(p, d') + \sum_{s\in N(p)\setminus q} m^{t-1}_{s\to p}(d'), \tag{3.12}$$

where $N(p) \setminus q$ indicates the neighbors of p other than q. $s(d, d^{'})$ is the smoothness function which represents that the agreement degree pixel p 's disparity with pixel q 's disparity. $m(p, d^{'})$ is the pixel dissimilarity of $d^{'}$, and $\sum_{s \in N(p) \setminus q} m_{s \to p}^{t-1}(d^{'})$ indicates the messages about disparity $d^{'}$ in previous iterations.

After certain iterations a final belief disparity is computed for each pixel by:

$$d^{*} = \arg\min_{d^{'}}(m(p, d^{'}) + \sum_{s \in N(p)} m_{s \to p}^{T}(d^{'})). \tag{3.13}$$

In Klaus' scheme, they used Loopy belief propagation proposed by Felzenszwalb [10], but modified it on segment level. For each segment, its disparity may be affected by interacting the message to its neighborhood. In this way, some segments which have wrong fitting results in previous step can be adjusted by its reliable neighborhood. The energy function is given as:

$$E(d) = E_{data}(d) + \lambda * E_{smooth}(d), \tag{3.14}$$

The discontinuity penalty λ in Klaus' scheme is defined as the product of two elements: the common border lengths between adjacent segments and the mean color similarity between adjacent segments [50]. The border length function is defined by the number of pixels belonging to one segment s_1 and its connective segments s_2 at the same time. The color similarity function measures the color similarity of segments s_1 and s_2. It assumes that segments with similar color are more likely to be assigned to the same disparity value. The function is defined as follows:

$$colorsimilarity(s_1, s_2) = (1 - \frac{\min(|meancolor(s_1) - meancolor(s_2)|, 255)}{255}) * 0.5 + 0.5, \tag{3.15}$$

where meancolour (s) indicates the component-wise summed up RGB values of pixels inside segment s divided by their number. In this function, the cost of assigning two neighboring segments of similar color with difference disparity values is higher than separated two segments of low similar color.

In our implementation, due to the truncation value set in plane fitting step, some segments has no plane fitted disparities. In belief propagation, those regions will be "filled" by the disparity plane from their reliable neighborhood. The implementation results are shown in Fig.3.9.

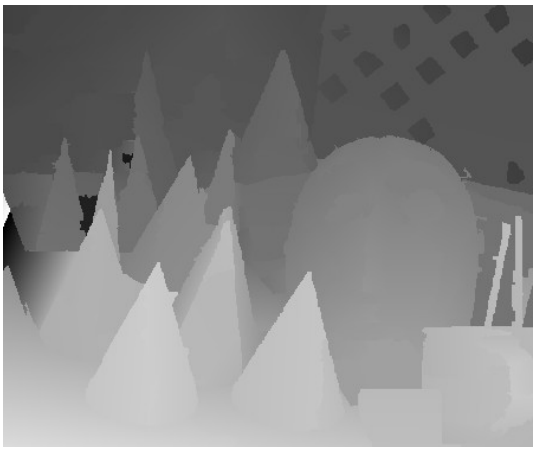
(a) Result of plane fitting.

(b) Result of belief propagation.

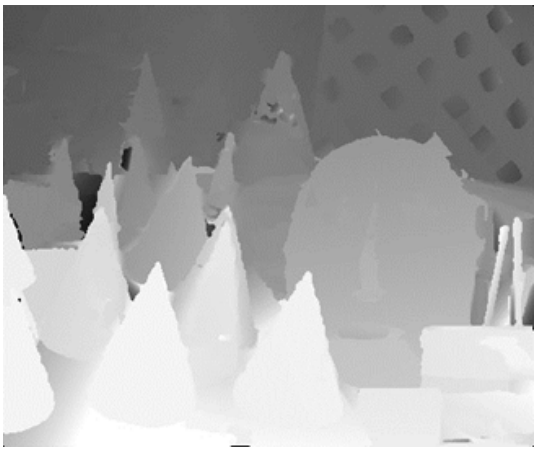
Figure 3.9: Disparity map after belief propagation.

3.5 The implementation results of Klaus' scheme for stereo matching

The four steps described above have been compiled to one project on Microsoft Visual Studio Platform. By making a comparison with Klaus' results on Middlebury, the implementation results are comparable and satisfactory. Especially for "Tsukuba", our implementation generates better contour of camera tri-pod which is not visible in Klaus' results. The entire final results are shown in Fig.3.10.



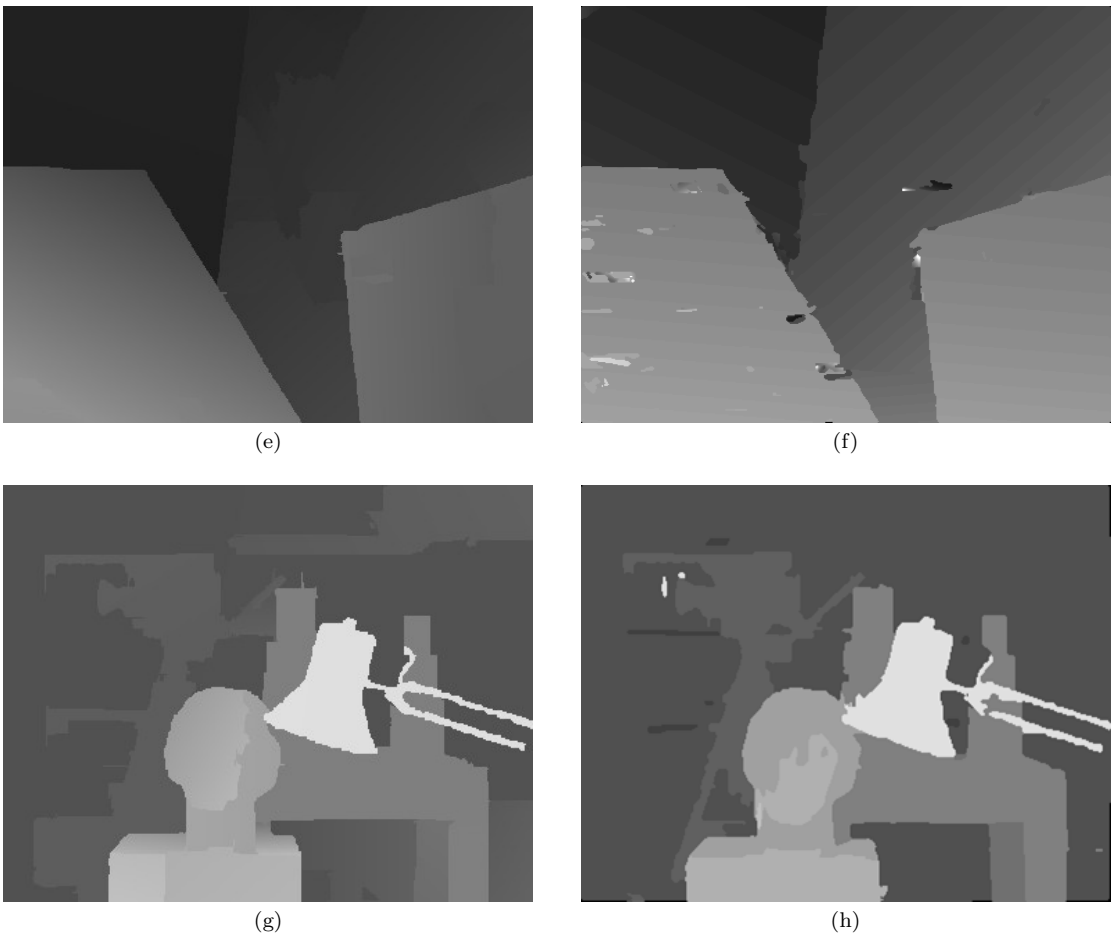
(a)



(b)

(c)

(d)



(e) (f)

(g) (h)

Figure 3.10: The comparison of our implementation and Klaus'. (a), (c), (e) and (g) are the results of Klaus'; (b), (d), (f) and (h) are our results.

Chapter 4

Segmentation based 2D-to-3D video conversion using optical flow algorithm

Based on the implemented framework of stereo matching, we propose a segmentation based 2D-to-3D video conversion method using high quality motion information acquired from optical flow calculation. Our scheme consists of depth generation based on motion information, mean-shift color segmentation and least square plane estimation. The basic construct is shown in Fig.4.1.

4.1 Original depth map generation based on optical flow algorithm with initial values

For 2D-to-3D video conversion, depth map generation is the most important step. In our scheme, depth map is generated completely based on motion information. Motion

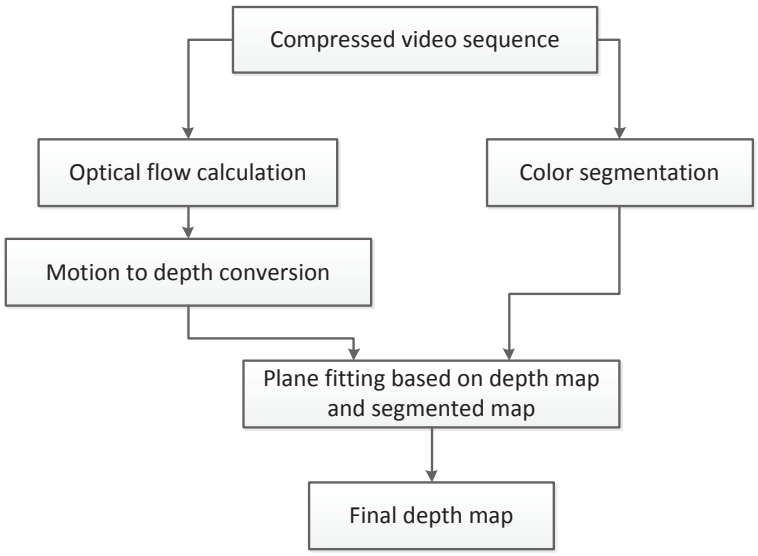


Figure 4.1: The flowchart of proposed segmentation based 2D-to-3D video conversion using optical flow algorithm.

information in this thesis means motion parallax, which refers to the relative motion between the viewing camera and the observed scene. For a moving observer, nearer object moves faster than the farther ones do. In this case, motion parallax provides an important cue to make depth perception.

We should notice that, as we have mentioned in Chapter 2, not all the video sequences can provide motion parallax to the depth, only those captured by a moving camera have motion parallax. In this case, we only focus on videos that have no or few IMOs (Independent Moving Objects) in them. In our scheme, the optical flow algorithm proposed by Brox [51] is used to get the motion parallax.

4.1.1 Brox model of optical flow algorithm

Optical flow method, as apparent motion of the image brightness pattern, aims to calculate the motion between two consecutive frames at time t and $t+1$. This method is

based on Taylor series using partial derivatives with respect to the spatial and temporal coordinates.

Optical flow method keeps the constraint that the brightness of moving object remains constant. This constraint can be described as follows:

$$I(x, y, t) = I(x + u, y + v, t + 1), \tag{4.1}$$

where $I(\Omega \rightarrow R_3)$ donates an image sequence; $\omega = (u; v; 1)^T$ is the motion vector between an image at time t and another image at time $(t + 1)$.

Assuming that the movement to be very small, for formula (4.1) by using Taylor series and further developing, we can get:

$$I_x u + I_y v + I_t = 0, \tag{4.2}$$

This is the linear version of the constancy assumption. However, the limitation of this assumption is that it is not suitable when a large displacement or movement happens. Brox provided a non-linearized model based on the assumptions of brightness constancy, gradient constancy, and discontinuity preserving spatia-temporal smoothness constraint. An energy minimization problem derived from these assumptions is as follows:

$$E(u, v) = E_{data} + \alpha E_{smooth}, \tag{4.3}$$

where the data term of the energy function is:

$$\begin{aligned} E_{data}(u, v) = \int_{\Omega} (|I(x + w) - I(x)|^2 + \\ \gamma |\nabla I(x + w) - \nabla I(x)|^2) dx, \end{aligned} \tag{4.4}$$

where γ is a weight factor, $x = (x; y; t)^T$ and $w = (u; v; 1)^T$. The smoothness term is:

$$E_{smooth}(u, v) = \int_{\Omega} \Psi(|\nabla_3 u|^2 + |\nabla_3 v|^2) dx, \tag{4.5}$$

where ∇_3 is the spatia-temporal gradient.

The $E(u, v)$ is highly nonlinear. In order to give better readability, abbreviations followed are defined.

$$\begin{aligned}
I_x &:= \partial_x I(x + w), \\
I_y &:= \partial_y I(x + w), \\
I_z &:= I(x + w) - I(x), \\
I_{xx} &:= \partial_{xx} I(x + w), \\
I_{xy} &:= \partial_{xy} I(x + w), \\
I_{yy} &:= \partial_{yy} I(x + w), \\
I_{xz} &:= \partial_x I(x + w) - \partial_x I(x), \\
I_{yz} &:= \partial_y I(x + w) - \partial_y I(x),
\end{aligned} \tag{4.6}$$

where the use of z instead of t emphasizes that the expression is not a temporal derivative but a difference that is sought to be minimized. And the minimization of this energy function is to solve the Euler-Lagrange equations as follows:

$$\begin{aligned}
&\Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_x I_z + \gamma(I_{xx} I_{xz} + I_{xy} I_{yz})) \\
&\qquad - \alpha div(\Psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2) \nabla_3 u) = 0,
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
&\Psi'(I_z^2 + \gamma(I_{xz}^2 + I_{yz}^2)) \cdot (I_y I_z + \gamma(I_{yy} I_{yz} + I_{xy} I_{xz})) \\
&\qquad - \alpha div(\Psi'(|\nabla_3 u|^2 + |\nabla_3 v|^2) \nabla_3 u) = 0,
\end{aligned} \tag{4.8}$$

with reflecting boundaries conditions.

Brox's scheme solves the non-linearized problem of optical flow and provides much more accurate results. Besides, this scheme is robust under noise, which is important to our scheme. Since most of the videos are compressed, and the noise may be introduced during the compressing process. The noise-robust method can give a guarantee to a better result of depth map from motion vectors.

After we get the motion parallax map, the depth values can be estimated according to the magnitude of the motion vectors. By making this conversion, the depth map is generated from the motion parallax. The conversion function is defined as follows:

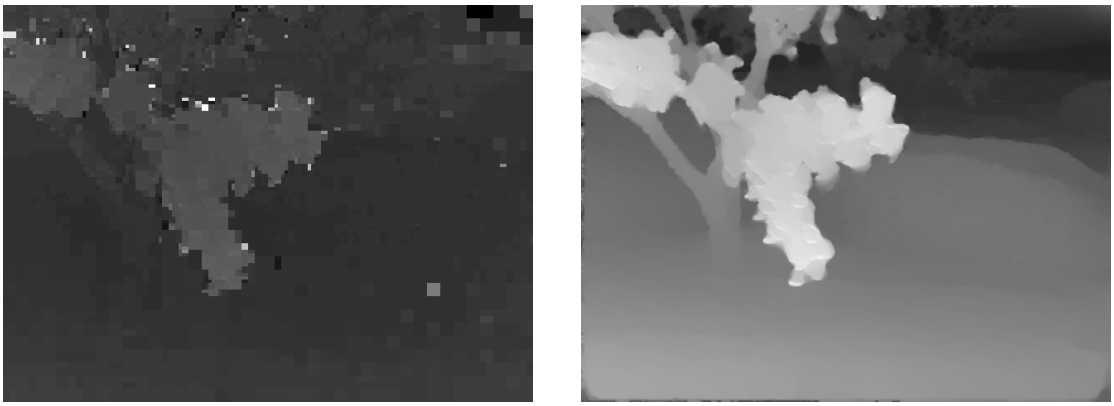
$$D(i,j) = \sqrt{u^2 + v^2}. \tag{4.9}$$

4.1.2 The initial values from compressed video

Although optical flow calculation can provide more accurate motion vectors compared to extract motion vectors from compressed video directly, the computational complexity is high. In order to acquire accurate motion vectors with better efficiency, we take the motion vectors from compressed videos as initial values for the optical flow calculation.

We modified the open source code JM8.6 based on our requirement to extract motion map from compressed video. From Fig.4.2 we can see that motion vectors from compressed video can provide sparse motion field. Since Brox scheme is a pyramid based scheme, and on each layers of pyramid, there are outer and inter iterative calculation to make convergence. Basically, the iterative progress is a coarse-to-fine progress, and each iteration step is to update motion vectors. In this case, the relative accurate initial values will consequently reduce the number of iterations, and to some extent improve the accuracy of the final results.

The comparison of the number of iterations and motion vectors' accuracy for the



(a) Motion map from compressed video. (b) Motion map from optical flow calculation.

Figure 4.2: Using motion vectors from compressed video as initial input for optical flow calculation.

test sequence with and without initial values are shown in Table 4.1. We take the sequence "Grove2" from optical flow benchmark [52] to make the test. In Table 4.1, "with" means initial values are motion vectors from compressed video; "without" means the initial values are zero; "AE" indicates the average angel error; "ED" indicates the end point error.

Table 4.1: The error analysis of motion vectors to Grove2.

Level	AE analysis		ED analysis	
	with	without	with	without
6	3.51	6.84	0.25	1.01
9	3.26	6.29	0.23	0.85
14	3.17	4.72	0.22	0.55
20	3.11	3.74	0.22	0.42
30	3.03	3.25	0.21	0.29

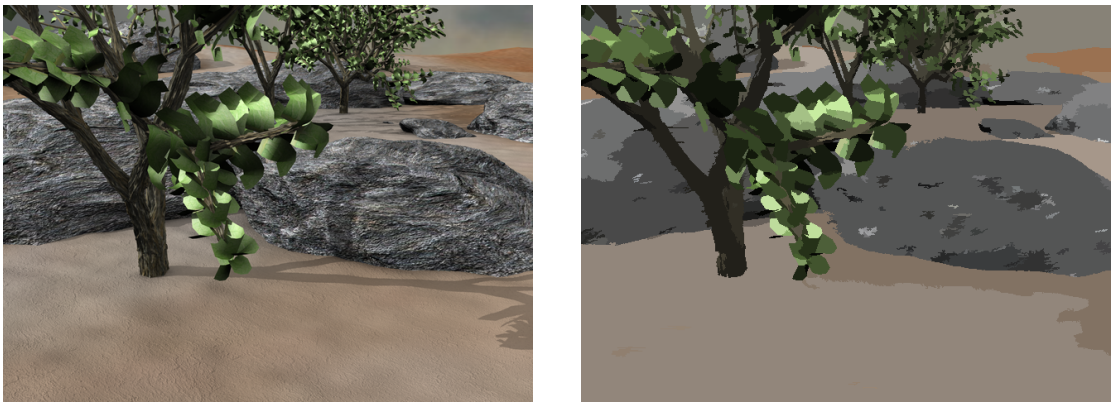
Table 4.1 shows the error analysis for different pyramid's layers calculation. Brox's algorithm is a pyramid based coarse-to-fine algorithm. In his scheme, each pyramid layer's size is calculated according to the input frame's size and the layer's number. In

our test, we set different layers and compute the final "AE" and "ED" for the tested sequence with and without using initial values. From the results we can see, it is clear that by using the initial values from compressed video, it only needs 9 levels with initial values to get the motion vectors as accurate as the 30 levels without using initial values. In this case, initial values from compressed video successfully reduce the number of levels, and consequently improve the efficiency of optical flow calculation.

4.2 Color segmentation to reference frame

Object boundaries information is very important to the depth map. However, due to the occlusion of objects and camera motion, the objects in depth map produced by optical flow do not have clear boundaries. To solve this problem, the mean shift color segmentation is used in our scheme.

Mean shift color segmentation is part of the framework we implemented in stereo matching. It is a popular method to separate the overlapped objects to get the boundaries. However, since this segmentation is based on color information, those videos which have natural views like the forests, the lawns, and the colorful gardens will not suitable to be dealt with. Because the natural color from those views will be affected by light, haze or other natural factors which are too complicated to be segmented accurately. In this case, the proposed scheme will produce much better results when it deals with the videos which have geometrical and relative complete objects in it. Fig.4.3 shows the color segmentation results to one frame of video.



(a) The original frame. (b) Color segmented frame.

Figure 4.3: Color segmentation to the reference frame.

4.3 Plane fitting within the homogenous region

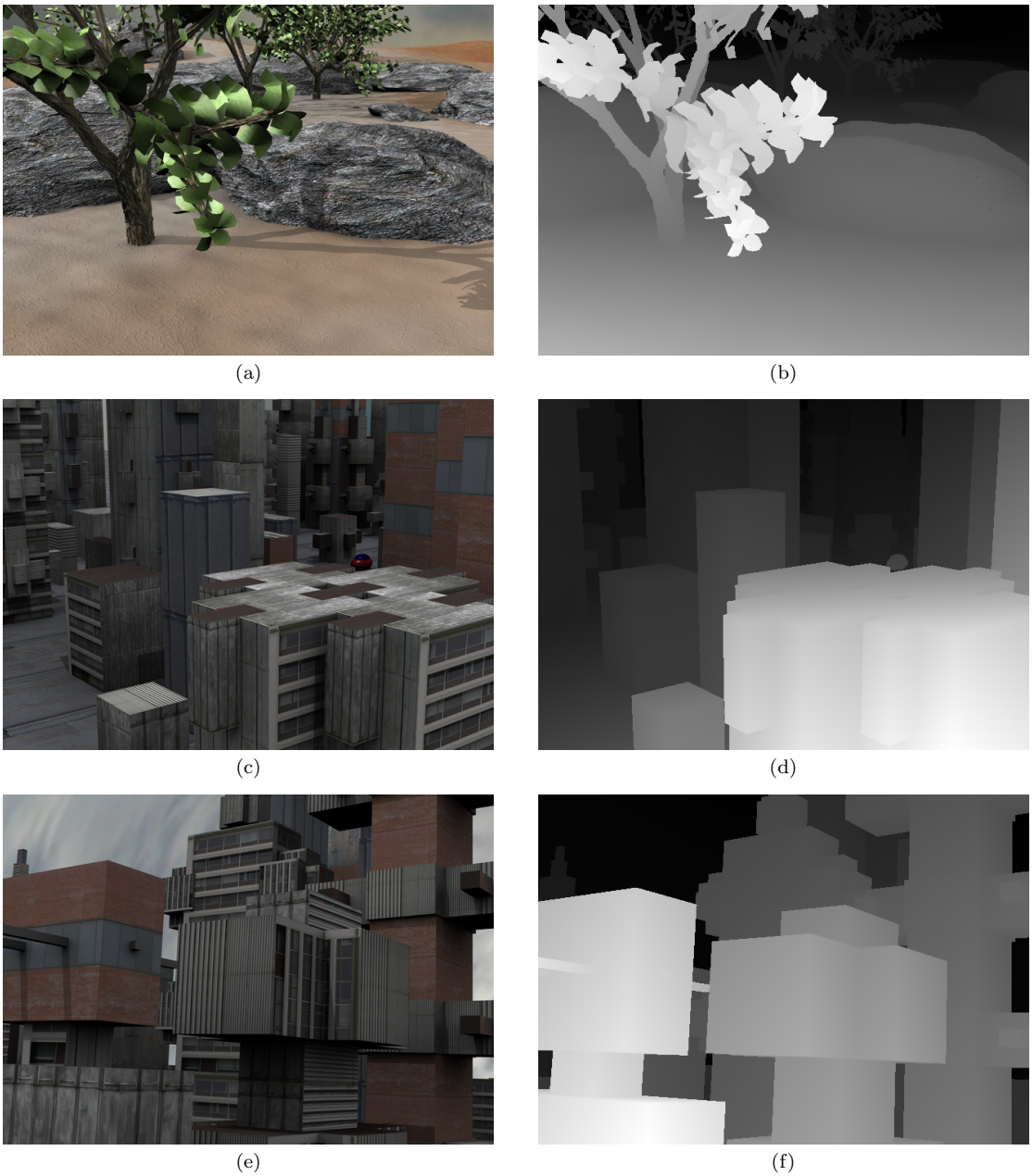
Plane fitting is a way to represent the disparity information in each segment with three parameters and normally it will come with the segmentation. It is worked upon the segmented regions, using the disparity values obtained from the motion parallax to fit disparity planes.

Plane fitting is also a part of the framework we have implemented in stereo matching. In our 2D-to-3D video conversion scheme, we still use the same function to make the plane estimation. Based on the generated depth values from optical flow calculation and the regions those depth values belong to, we can calculate the parameters to represent the depth information of one region. In this way, some outliers due to the occlusion and other factors, especially located at the boundaries will be displaced by more reliable values.

Chapter 5

Experimental results

In our experiments, we first use three image sequences with ground truth map correspondingly from [52] to test the efficiency of motion extraction part with and without initial values, and take endpoint error and angular error to measure motion vectors' accuracy. They are "Grove2" (640×480), "Urban2" (640×480) and "Urban3" (640×480). Then we add four more video sequences captured from real world without ground truth to test our proposed segmentation based 2D-to-3D video conversion using motion information method. They are "Flowers" (352×288) from the video trace library [53], "Horse" (270×480), "Yoki" (640×480) from [54], and a documentary named "the Imperial Palace1" (1280×720). In the third part, we use DIBR (Depth Image Based Rendering) technique to synthesis the 3D image. Fig. 5.1 shows the original frames with ground truth, Fig. 5.2 shows the original frames without ground truth.



(a) (b) (c) (d) (e) (f)

Figure 5.1: Original frames from benchmark and their corresponding ground truth. (From top to bottom: Grove2, Urban2, Urban3.)

(a) (b)

(c) (d)

Figure 5.2: Original video frames. (From top to bottom: Horse, Flower, Palace and Yoki.)

5.1 Motion vector generated from optical flow with initial values from compressed video

In this section, we test three image sequences using ground truth from optical flow benchmark to analyse the motion vectors' accuracy, and make the comparison between the motion vectors with and without initial values from compressed videos. The pyramid levels are calculated based on the input frame's size using the following function:

$$levels = \left\lceil \lg(\frac{20}{\min(ht, wt)} - param) \right\rceil, \tag{5.1}$$

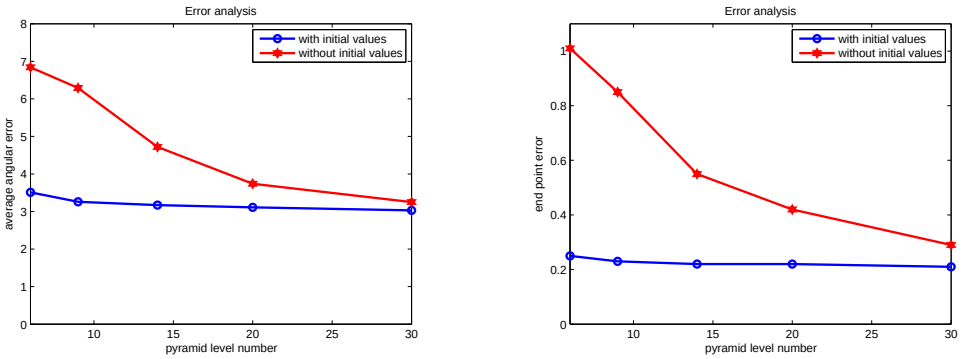
where (ht, wt) is the height and width of the frame, $param$ is the parameter that can be adjusted to produce different kinds of pyramid levels.

For efficiency, we compare the iteration numbers on each pyramid level of optical flow calculation. For accuracy, we take "AE" (which indicates the average angular error) and "ED" (which indicates the end point error) to make the measurement.

Table 5.1: The error analysis of motion vectors to Grove2.

Level	AE analysis		ED analysis	
	with	without	with	without
6	3.51	6.84	0.25	1.01
9	3.26	6.29	0.23	0.85
14	3.17	4.72	0.22	0.55
20	3.11	3.74	0.22	0.42
30	3.03	3.25	0.21	0.29

The Table 5.1, Table 5.2 and Table 5.3 and plot figures below are showing the "AE" and "ED" comparison for the tested three image sequences. In the table, "with" means initial values are motion vectors from compressed video; "without" means the initial



(a) "AE" analysis for "Grove2". (b) "ED" analysis for "Grove2".

Figure 5.3: The error analysis "AE" and "ED" for motion extraction of "Grove2" with and without initial values.

Table 5.2: The error analysis of motion vectors to Urban3.

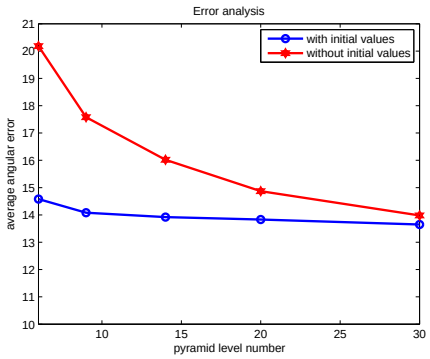
Level	AE analysis		ED analysis	
	with	without	with	without
6	14.58	20.18	1.57	3.82
9	14.08	17.58	1.54	3.18
14	13.92	16.02	1.50	2.52
20	13.83	14.87	1.45	2.16
30	13.65	13.98	1.37	1.83

Table 5.3: The error analysis of motion vectors to Urban2.

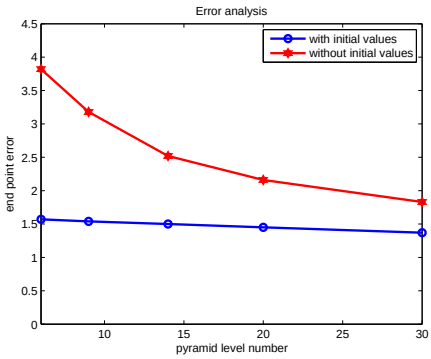
Level	AE analysis		ED analysis	
	with	without	with	without
6	4.70	11.58	0.94	4.99
9	4.66	9.50	0.95	3.80
14	4.44	8.04	0.94	2.58
20	4.35	7.14	0.94	2.09
30	4.21	5.96	0.93	1.60

values are zero.

In our experiments, we produced five different pyramid levels 6, 9, 14, 20, and

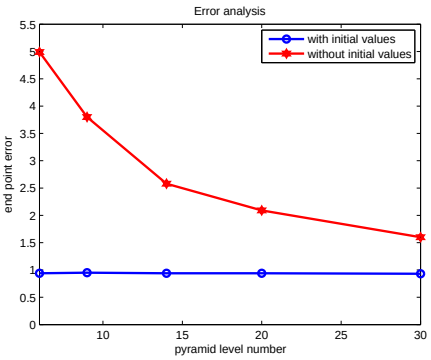


(a) "AE" analysis for "Urban3".

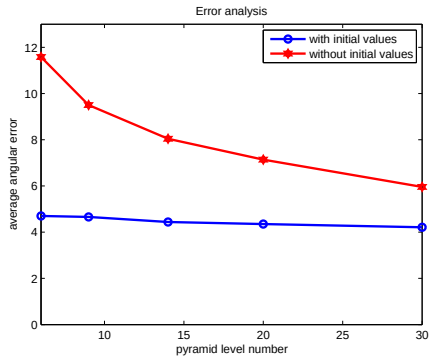


(b) "ED" analysis for "Urban3".

Figure 5.4: The error analysis "AE" and "ED" for motion extraction of "Urban3"with and without initial values.



(a) "AE" analysis for "Urban2".



(b) "ED" analysis for "Urban2".

Figure 5.5: The error analysis "AE" and "ED" for motion extraction of "Urban2"with and without initial values.

30, respectively, to test the accuracy and efficiency of proposed scheme in different situations. From the tables and plot figures we can see that the accuracy of the motion vectors are improved by adding the initial values from compressed video. The values of "AE" and "ED" keep stable with the initial values and much lower than the values without initial values. Besides, the improvements are obvious when the pyramid level numbers' are low, which mean we can get almost the same accuracy results with fewer

pyramid layer number. In this case, we can improve the optical flow calculation's efficiency from another aspect.

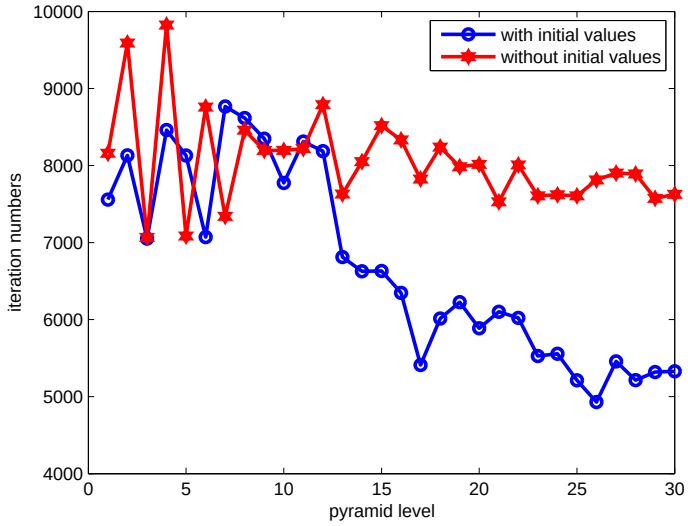


Figure 5.6: The comparison of iteration numbers between optical flow with and without initial values for "Urban3".

Fig. 5.6, Fig. 5.7 and Fig. 5.8 shows the iteration numbers of optical flow calculation. We set the total pyramid levels as 31, because 31 is the largest pyramid levels in our experiments based on input frame's size within the reasonable range. The x coordinate is each pyramid level, and the y coordinate is the iteration numbers.

Since Brox's scheme is based on the layered pyramid, and for each level, there are numbers of iterations to calculate until the convergence. Those figures illustrate that the initial values can reduce the number of iterations for each level. From the figures we can see, the decrements are not so obvious for the bottom several layers since the bottom layers have relative lower resolution and the motion vectors to be updated by initial values are not so much. However, for the upper layers, the iteration numbers decrease a lot, and in that case, the processing speed is improved.

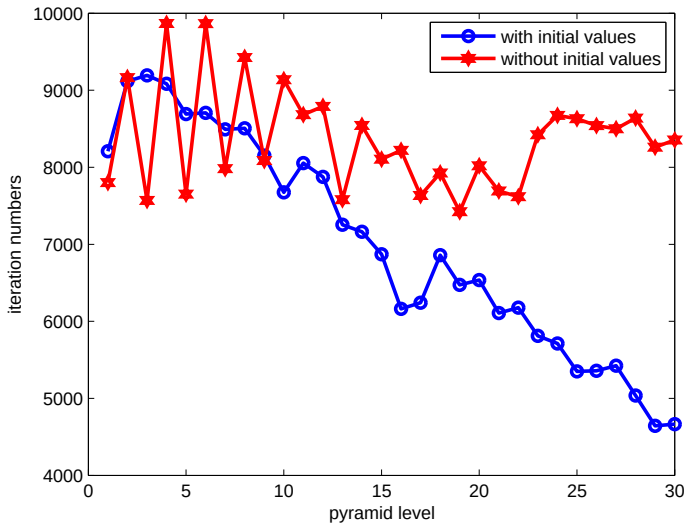


Figure 5.7: The comparison of iteration numbers between optical flow with and without initial values for "Urban2".

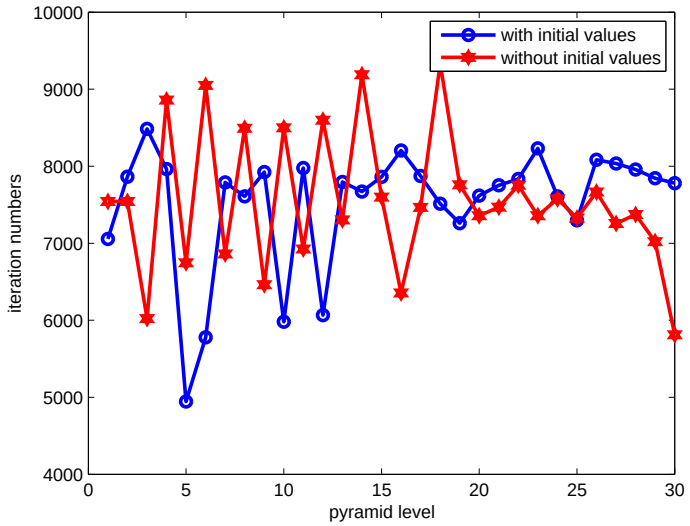
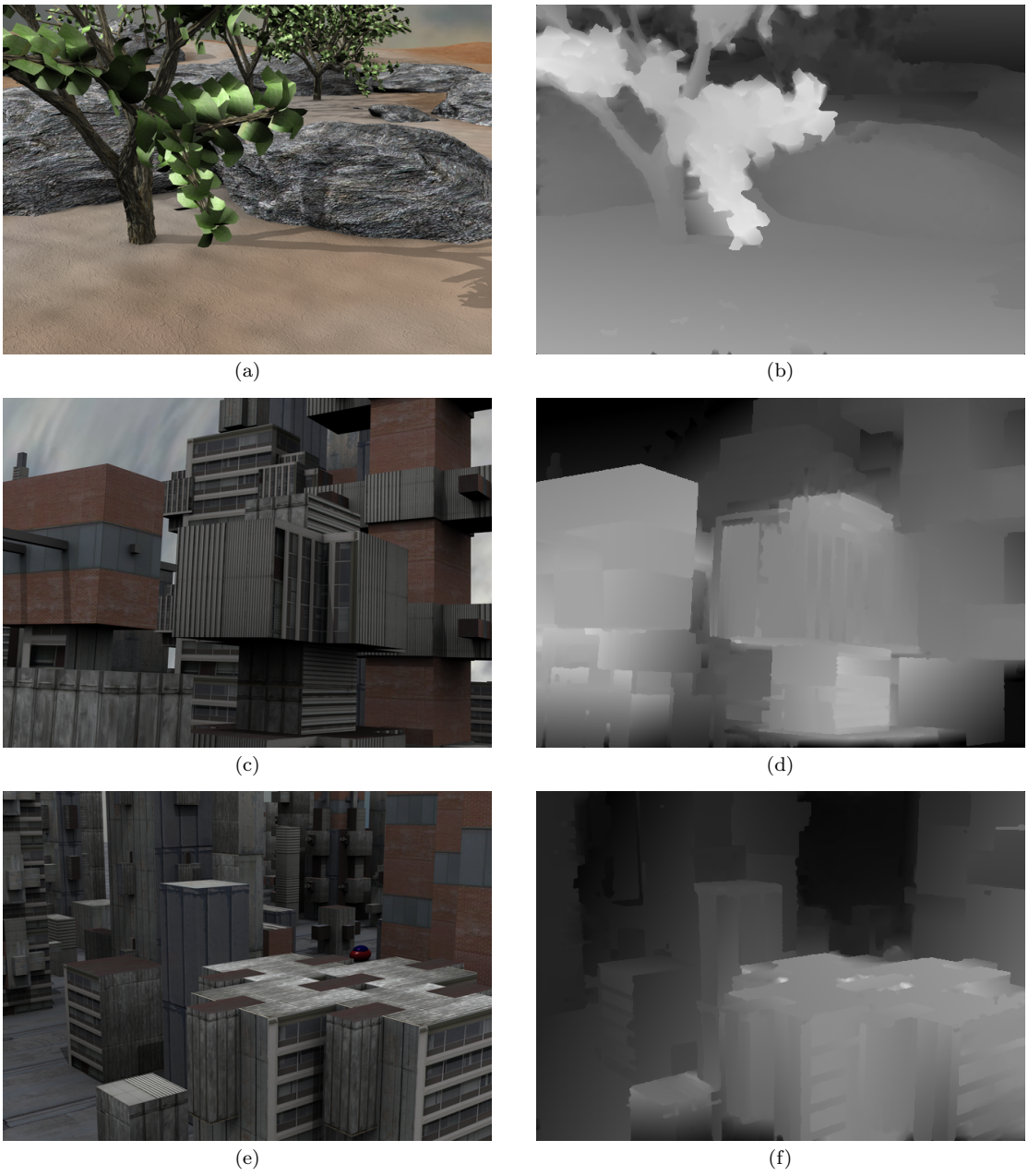


Figure 5.8: The comparison of iteration numbers between optical flow with and without initial values for "Grove2".

5.2 The depth map generated from video sequence

In this section, the tested image sequences and their corresponding depth map are shown (Fig. 5.9, Fig. 5.10). From the results we can see, the proposed scheme successfully produces sharp object boundaries.

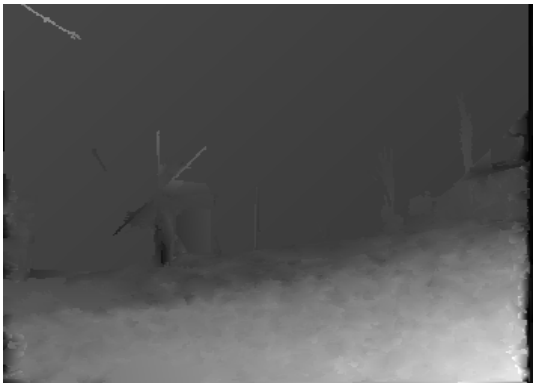
Fig. 5.12 gives the detailed pictures of raw depth maps produced by optical flow and the depth maps post-processed by color segmentation and plane fitting. By comparing the contours of the objects, especially the leaves of "Grove2", the building shape of "Urban2" and "Urban3", the windmill of the "Flowers", the head of "Horse", the body shape "Yoki", and the small sculptures on the roof of "the Imperial Palace", it is clear that color segmentation can give much more accurate boundaries.



(a) (b) (c) (d) (e) (f)

Figure 5.9: Original frames and their corresponding depth maps. From top to bottom: Grove2, Urban3, and Urban2.

(a)

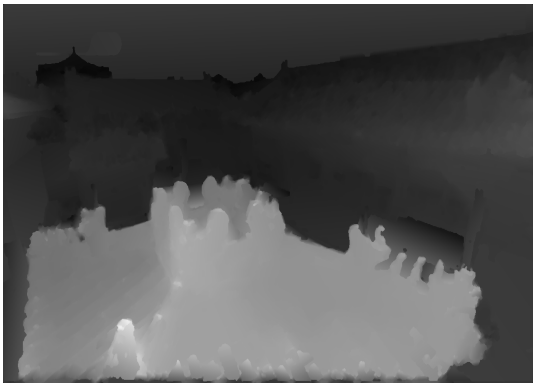


(b)

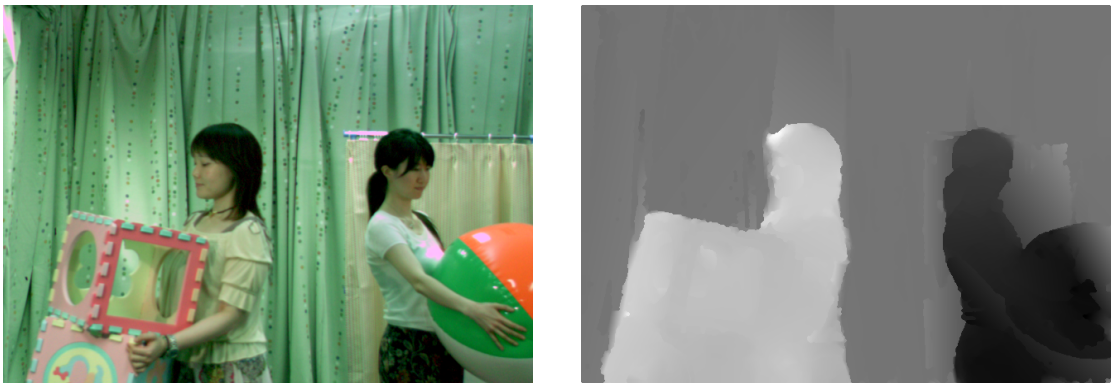
(c)

(d)

(e)

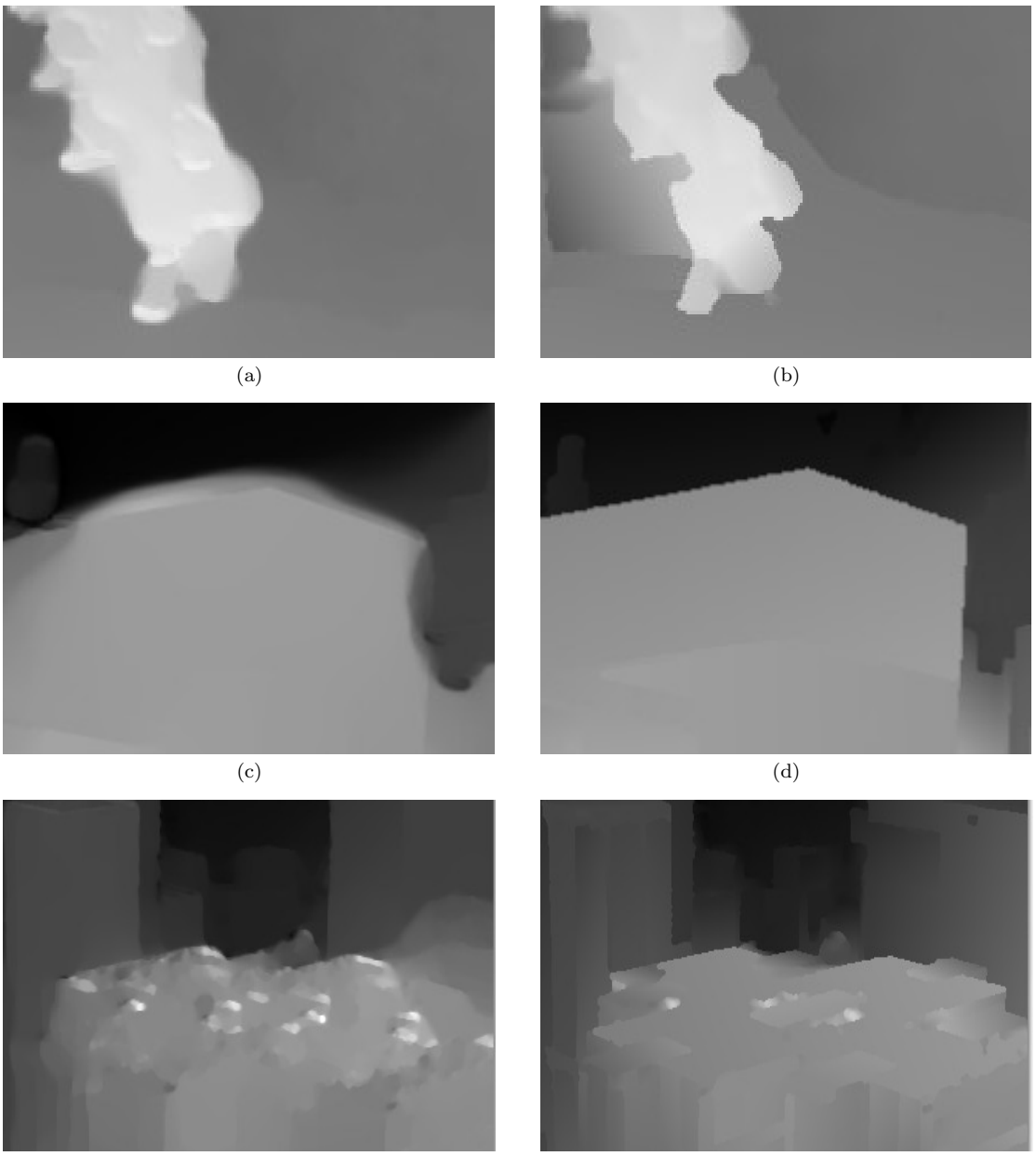


(f)



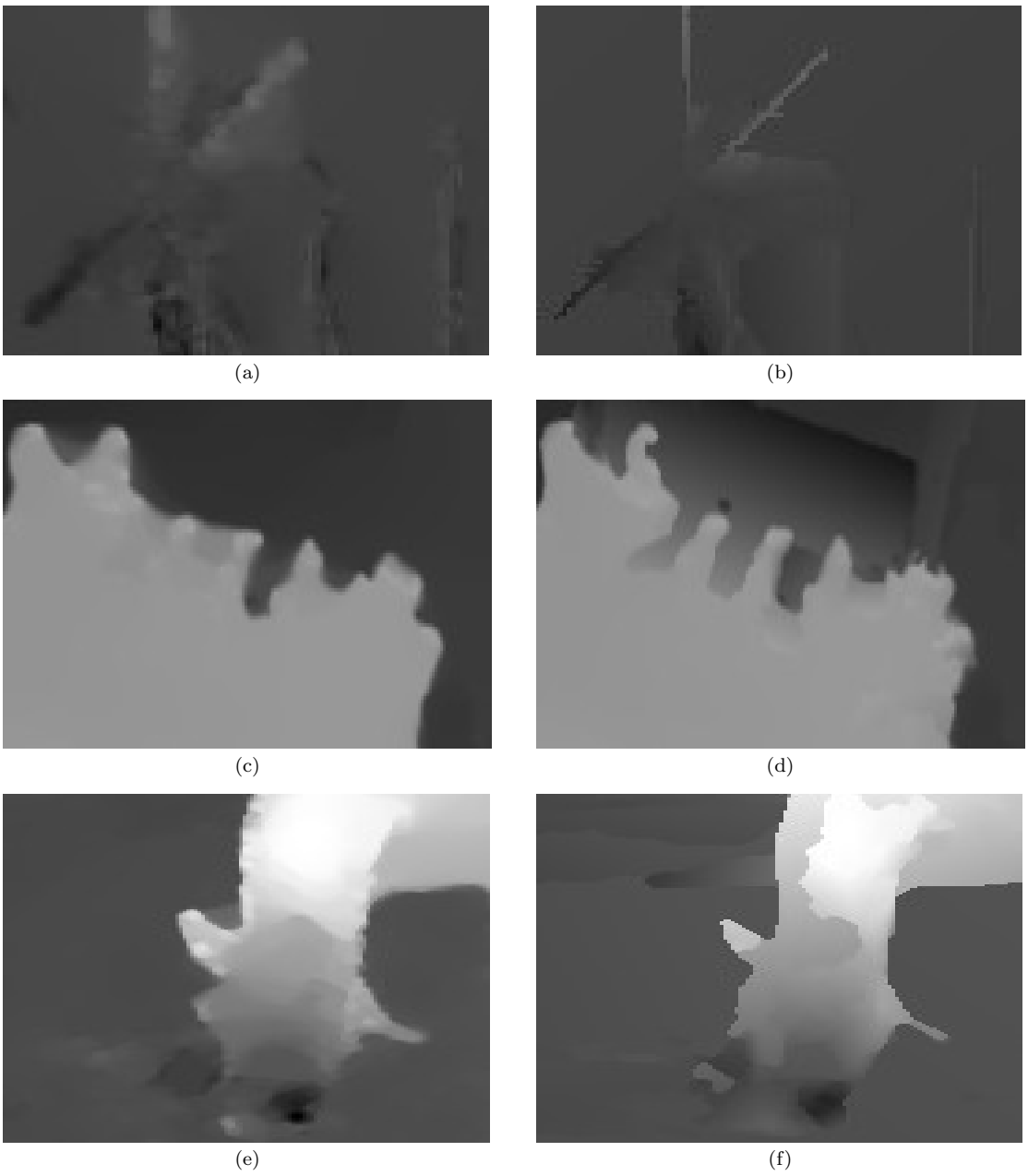
(g) (h)

Figure 5.10: Original frames and their corresponding depth maps. From top to bottom: Flower, Horse, Palace, and Yoki.

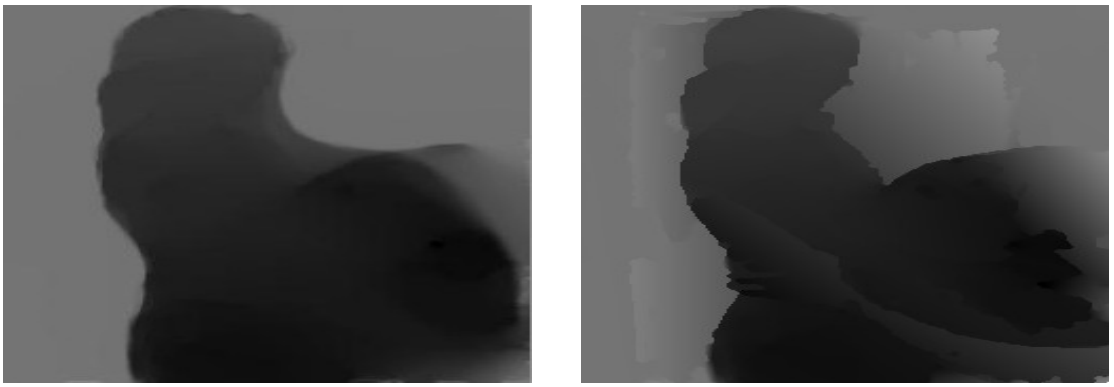


(a) (b) (c) (d) (e) (f)

Figure 5.11: The patches from depth produced by optical flow and after segmentation and plane fitting (left column is the optical flow produced patches, right column is patches after segmentation and plane fitting).

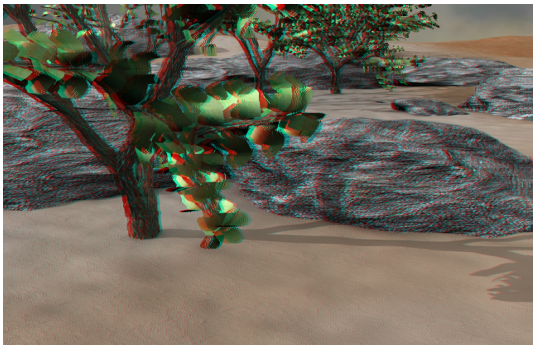


(a) (b) (c) (d) (e) (f)

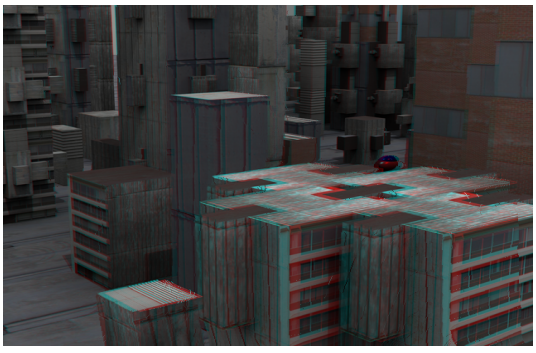


(g) (h)

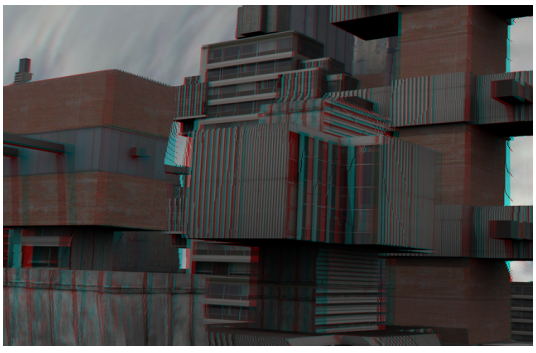
Figure 5.12: The patches from depth produced by optical flow and after segmentation and plane fitting (left column is the optical flow produced patches, right column is patches after segmentation and plane fitting).



(a)



(b)



(c)

(d) (e)

(f) (g)

Figure 5.13: The DIBR results of the tested image sequences.

5.3 The DIBR results

To generate the stereoscopic 3D video, DIBR is used to synthesize the left-eye view video based on the estimated depth map and monoscopic video input [55]. The DIBR results are shown in Figure 5.13. The 3D visual feelings can be achieved by wearing red and green glasses.

5.4 Summary

In this chapter, we tested our proposed 2D-to-3D video conversion scheme by using the image sequences with ground truth from optical flow benchmark and the videos without ground truth from other sources. We separated the experiments as two parts: in the first part, we analysed the accuracy and efficiency of our scheme by taking the "AE" and "ED" measurements and iteration numbers between optical flow calculation with and without initial values from compressed videos. Three image sequences were tested in this part, and from the results we can see clearly that the accuracy of optical flow calculation improved a lot by adding the initial values, especially when the pyramid's layers are low. And the efficiency of calculation is also improved, since the iteration numbers reduced on upper pyramid levels with initial values.

Besides, we tested another four image sequences captured from the real world to extend the practicality of our scheme. The experimental results met our expectations, especially for the object boundaries. We gave the detail figures to make the comparison between the disparity map acquired from the motion vectors directly using optical flow calculation and the disparity map using our proposed segmentation based method. The results were obvious that our method made the boundaries of objects more clear and sharper. In this way, our proposed scheme successfully produced good results for the videos with few independent moving objects in it.

Chapter 6

Conclusions and future work

3D representation has rapidly occupied the market of ultimate visual experience, and the most important step in 3D system is the depth map generation. Thousands of algorithms have been proposed to extract the depth from binocular image pairs or monocular image sequences. In this thesis, a color segmentation based 2D-to-3D video conversion using high quality motion information scheme has been proposed. This scheme was based on the well-known color segmentation frame work for stereo matching proposed by Klaus.

One of the important challenges for depth generation is how to keep the object boundaries of extracted depth map well. From this aspect, color segmentation based methods provide a good solution. Based on the assumption that the content of image can be constructed by a group of non-overlapping planes in the depth space, each plane is supposed to contain at least one homogeneous color region, and the discontinuity only occurs on region boundaries, color segmentation separates the objects in image and holds the edges information more accurately. In our scheme, we kept the color segmentation and plane estimation steps from Klaus' stereo matching scheme and employed

the optical flow algorithm to obtain the motion parallax for depth map. Optical flow algorithm, as a typical method to acquire motion information, has a better performance compared to the other method. The optical flow algorithm we used was originally from Brox, and we made some adjustment by using the initial values to improve the efficiency and accuracy at the same time. Initial values from compressed video were extracted conveniently as the input for optical flow calculation part. This adjustment could make the convergence be reached within fewer iteration times.

The experimental results showed that our proposed method successfully produced sharp object boundaries. From the detail figures we can see that the color segmentation indeed retained the edges details in good performance. By making the iteration times comparison between optical flow algorithm with and without initial values, we can see that the good initial values from compressed video took less time than that without initial values, and to some extent improved the method efficiency.

However, the limitations of our scheme still exist: due to the optical flow calculation, the computational complexity is higher than other method to acquire the motion information such as getting the motion vectors directly from compressed video. Besides, our method is limited to the video sequences without complex objects moving in it, since that our method is based on the motion parallax, and the moving objects should keep consistency to some extent. What's more, the color-segmentation based method is hard to modularize, which mean this kind of method is hard to implement on GPU, and will not reach real-time processing.

It has been noticed on July 18, 2012 that an independent and similar work has been conducted by another research group [56]. In their scheme, they use a classical constrained optical flow method [57] to get motion vectors, combined with mean shift color segmentation to generate final depth map. The motion based depth map and the

segmented map are integrated into one depth map using breadth-first search method. Their results are impressive, but they did not give the experimental results for practical image sequences. In comparison to their scheme, although we have similar framework, there are still some differences. Our scheme employs a different optical flow method and makes an improvement by using the initial values from compressed videos. Besides, we use different plane estimation methods to get the final depth map. However, We should admit that the optical flow method they used produces more accurate motion map than ours. But the final results are comparable after post-processing with color segmentation and plane estimation.

More work will be done in our future research. We are trying to construct a hybrid model which can deal with more complex video sequences. Pictorial cues like texture gradient, linear perspective are still very important cues for 2D to 3D video conversion, and how to utilize the motion information and pictorial cues to their full extent is still the key point in future.

References

[1] C. Wheatstone, "Contributions to the physiology of vision. c part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision," *Philosophical Transactions of the Royal Society of London*, vol. 128, pp. 371 – 394, January 1838.

[2] L. Tang, M. Garvin, K. Lee, W. Alward, Y. Kwon, and M. Abramoff, "Robust multiscale stereo matching from fundus images with radiometric differences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2245 – 2258, November 2011.

[3] L. Zhang, C. Vazquez, and S. Knor, "3D-TV content creation: automatic 2D-to-3D video conversion," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 372–393, June 2011.

[4] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *The 18th International Conference on Pattern Recognition*, vol. 3, Hong Kong, China, August 20-24, 2006, pp. 15–18.

[5] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *IEEE Conference on Stereo and*

Multi-Baseline Vision, Kauai, HI, United States, December 9-10, 2001, pp. 131–140.

[6] P. Foggia, J. M. Jolion, A. Limongiello, and M. Vento, "A new approach for stereo matching in autonomous mobile robot applications," in *The 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, January 6-12, 2007, pp. 2103–2108.

[7] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, United States, June 17-22, 2007, pp. 1–8.

[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, November 2001.

[9] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, February 2002.

[10] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Washington, DC, United States, June 27-July 2, 2004, pp. I–261 – I–268.

[11] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon, "Global stereo reconstruction under second order smoothness priors," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, United States, June 23-28, 2008, pp. 2570–2577.

[12] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *IEEE Conferenceon Computer Vision and Pattern Recognition*, Minneapolis, MN, United States, June 17-22, 2007, pp. 1784–1791.

[13] vision.middlebury.edu, http://vision.middlebury.edu/flow/, last visited on July 30, 2012.

[14] Z.Wang and Z. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, United States, June 23-28, 2008, pp. 1–8.

[15] M. Bleyer, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, United States, June 13-18, 2010, pp. 1570–1577.

[16] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo: joint stereo matching and object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, United States, June 20-25, 2011, pp. 3081 – 3088.

[17] K. Zhang, J. Lu, G. Lafruit, R. Lauwereins, and L. V. Gool, "Real-time and accurate stereo: a scalable approach with bitwise fast voting on CUDA," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, pp. 867–878, July 2011.

[18] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *IEEE Conference on Computer Vision Workshops*, Barcelona, Spain, November 6-13, 2011, pp. 467–474.

[19] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang, "Stereo matching with reliable disparity propagation," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, Hangzhou, China, May 16-19, 2011, pp. 132–139.

[20] J. Kowalczuk, E. Psota, and L. Perez, "Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–11, June 2012.

[21] I. Ideses, L. P. Yaroslavsky, and B. Fishbain, "Real-time 2D to 3D video conversion," *Journal of Real-Time Image Process*, vol. 2, no. 1, pp. 3–9, October 2007.

[22] L. Po, X. Xu, Y. Zhu, S. Zhang, K. Cheung, and C. Ting, "Automatic 2D-to-3D video conversion techique based on depth-from-motion and color segmentation," in *IEEE Conference on Signal Processing*, Beijing, China, October 24-28, 2010, pp. 1000–1003.

[23] M. T. Pourazad and P. Nasiopoulos, "An h.264-based scheme for 2D to 3D video conversion," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 742–748, May 2009.

[24] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, "Generating the depth map from the motion information of H.264-encoded 2D video sequence," *Journal of Image and Video Processing*, vol. 2010, no. 4, p. 13, January 2010.

[25] D. Min and K. Sohn, "A stereoscopic video generation method using stereoscopic display characterization and motion analysis," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 188–197, June 2008.

[26] Y. C. Lee, J. W. Choi, G. H. Lee, J. H.Park, and S. G. Lee, "The virtual time domain depth estimation of stereoscopic sequence using optical flow," in *IEEE Conference on Computers, Communications, Control and Power Engineering*, vol. 3, Beijing, China, October 28-31, 2002, pp. 1599–1602.

[27] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, April 1981.

[28] S. Yeong, J. H. Kim, J. H. Kim, D. W. Lee, and K.-R. Cho, "3D depth estimation for target region using optical flow and mean-shift algorithm," in *IEEE Conference on Control, Automation and Systems*, Seoul, Korea, October 14-17, 2008, pp. 34–39.

[29] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *The 7th International Joint Conference on Artificial Intelligence*, vol. 2, Vancouver, Canada, August 24-28, 1981, pp. 674–679.

[30] D. Comanicu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[31] C. Zhang, Z. Z. Yin, and D. Florencio, "Improving depth perception with motion parallax and its application in teleconferencing," in *IEEE Conference on Multimedia Signal Processing*, Rio De Janeiro, Brazil, October 5-7, 2009, pp. 1–6.

[32] J. Kim, Y. Choe, and Y. Kim, "High-quality 2D to 3D video conversion based on robust MRF-based object tracking and reliable graph-cut-based contour refinement," in *IEEE Conference on ICT Convergence*, Seoul, Korea, September 28-30, 2011, pp. 360–365.

[33] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Journals on Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 97–108, February 1993.

[34] R. Hariharan and A. Rajagopalan, "Shape-From-Focus by tensor voting," *IEEE Journals and Magazines on Image Processing*, vol. 21, no. 7, pp. 3323–3328, July 2012.

[35] J. Yang and D. Schonfeld, "Virtual focus and depth estimation from defocused video sequences," *IEEE Journals and Magazines on Image Processing*, vol. 19, no. 3, pp. 668–679, March 2010.

[36] F. Yu, J. Liu, Y. Ren, J. Sun, Y. Gao, and W. Liu, "Depth generation method for 2D to 3D conversion," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Antalya, Turkey, May 16-18, 2011, pp. 1–4.

[37] Y. Y. Ding, J. Yu, and P. Sturm, "Multiperspective stereo matching and volumetric reconstruction," in *IEEE Conference on Computer Vision*, Kyoto, Japan, September 29 - October 2, 2009, pp. 1827–1834.

[38] K. M. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, United States, June 20-25, 2009, pp. 1956–1963.

[39] C. Zou and J. Chen, "Recovering depth from a single image using dark channel prior," in *IEEE Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing*, London, United Kindom, June 9-11, 2010, pp. 93–96.

[40] T. Y. Kuo and Y. C. Lo, "Depth estimation from a monocular view of the outdoors," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 817–822, May 2011.

[41] W. H. Zhou, L. L. Lin, X. H. Wei, and B. Lou, "Combining dark channel prior and color cues for road following in outdoor environments," in *IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, Hong Kong, China, September 26-29, 2010, pp. 2197–2200.

[42] Z. B. Zhang, Y. Z. Wang, T. T. Jiang, and W. Gao, "Visual pertinent 2D-to-3D video conversion by multi-cue fusion," in *IEEE Conference on Image Processing*, Brussels, Belgium, September 11-14, 2011, pp. 909–912.

[43] Y. J. Jung, A. Baik, J. Kim, and D. Park, "A novel 2D-to-3D conversion technique based on relative height depth cue," in *Proceedings of SPIE on IS and T Electronic Imaging*, vol. 7237, 72371U, no. 1, San Jose, CA, United States, January 2009.

[44] G. Palou and P. Salembier, "Occlusion-based depth ordering on monocular images with binary partition tree," in *IEEE Conference on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 22-27, 2011, pp. 1093–1096.

[45] X. Q. Wang, L. Q. S. Y. Z. Jiang, and Z. Y. Zhang, "A new improved depth estimation algorithm based on gradient feature for preserving consistency among views," in *IEEE Conference on Multimedia Information Networking and Security*, Shanghai, China, November 4-6, 2011, pp. 37–40.

[46] G. S. Lin, C. Y. Yeh, W. C. Chen, and W. N. Lie, "A 2D to 3D conversion scheme based on depth cues analysis for MPEG videos," in *IEEE Conference on Multimedia and Expo*, Suntec City, Singapore, July 19-23, 2010, pp. 1141–1145.

[47] S. Battiatoa, S. Curtib, M. L. Casciac, M. Tortorac, and E. Scordatoc, "Depth-Map generation by image classification," in *Proceedings of SPIE Electronic Imaging - Three - Dimensional Image Capture and Applications*, vol. 5302, San Jose, CA, United States, January 2004, pp. 21–28.

[48] M. Pourazad, P. Nasiopoulos, and A. Bashashati, "Random forests-based 2D-to-3D video conversion," in *IEEE Conference on Electronics, Circuits, and Systems*, Athens, Greece, December 12-15, 2010, pp. 150–153.

[49] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Journals and Magazines on Information Theory*, vol. 21, no. 1, pp. 32–40, January 1975.

[50] M. Bleyer and M. Gelautz, "Graph-based surface reconstruction from stereo pairs using image segmentation," in *Proceedings of SPIE*, vol. 5665, San Jose, CA, United States, January 16-20, 2005, pp. 288–299.

[51] T.Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *The 8th European Conference on Computer Vision*, vol. 4, Prague, Czech Republic, May 11-14, 2004, pp. 25–36.

[52] vision.middlebury.edu, http://vision.middlebury.edu/flow/eval/, last visited on July 30, 2012.

[53] Video.Trace.Library, http://trace.kom.aau.dk, last visited on July 30, 2012.

[54] Mobile.3DTV, http://sp.cs.tut.fi/mobile3dtv/stereo-video/, last visited on July 30, 2012.

[55] W. J. Tam, f. Speranza, L. Zhang, R. Renaud, J. Chan, and C. Vazquez, "Depth image based rendering for multiview stereoscopic displays: Role of information at object boundaries," *SPIE Conference on Three-Dimensional TV, Video, and Display IV*, vol. 6016, pp. 75–85, November 2005.

[56] C. Liu and L. Christopher, "Depth map estimation from motion for 2D to 3D conversion," in *IEEE Conference on Electronics/Information Technology*, Indianapolis, IN, United States, May 6-8, 2012, pp. 1–4.

[57] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, United States, June 13-18, 2010, pp. 2432 – 2439.