VIETNAM NATIONAL UNIVERSITY, HANOI

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Dinh Trung Anh

DEPTH ESTIMATION FOR MULTI-VIEW VIDEO CODING

Major: Computer Science

HA NOI - 2015

VIETNAM NATIONAL UNIVERSITY, HANOI

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Dinh Trung Anh

DEPTH ESTIMATION FOR MULTI-VIEW VIDEO CODING

Major: Computer Science

Supervisor: Dr. Le Thanh Ha

Co-Supervisor: BSc. Nguyen Minh Duc

HA NOI – 2015

AUTHORSHIP

"I hereby declare that the work contained in this thesis is of my own and has not been previously submitted for a degree or diploma at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no materials previously published or written by another person except where due reference or acknowledgement is made."

Signature:……

SUPERVISOR'S APPROVAL

"I hereby approve that the thesis in its current form is ready for committee examination as a requirement for the Bachelor of Computer Science degree at the University of Engineering and Technology."

Signature:……

ACKNOWLEDGEMENT

Firstly, I would like to express my sincere gratitude to my advisers Dr. Le Thanh Ha of University of Engineering and Technology, Viet Nam National University, Hanoi and Bachelor Nguyen Minh Duc for their instructions, guidance and their research experiences.

Secondly, I am grateful to thank all the teachers of University of Engineering and Technology, VNU for their invaluable lessons which I have learnt during my university life.

I would like to also thank my friends in K56CA class, University of Engineering and Technology, VNU.

Last but not least, I greatly appreciate all the help and support that members of Human Machine Interaction Laboratory of University of Engineering and Technology and Kotani Laboratory of Japan Advanced Institute of Science and Technology gave me during this project.

Hanoi, May 8th, 2015

Dinh Trung Anh

ABSTRACT

With the advance of new technologies in the entertainment industry, the Free-Viewpoint television (TV), the next generation of 3D medium, is going to give users a completely new experience of watching TV as they can freely change their viewpoints. Future TV is going to not only show but also let users "live" inside the 3D scene. A simple approach for free viewpoint TV is to use current multi-view video technology, which uses a system of multiple cameras to capture the scene. The views at positions where there is a lack of camera viewpoints must be synthesized with the support of depth information. This thesis is to study Depth Estimation Reference Software (DERS) of Moving Pictures Expert Group (MPEG) which is a reference software for estimating depth from color videos captured by multi-view cameras. It also provides a method, which uses stored background information to improve the depth quality taken from the reference software. The experimental results exhibit the quality improvement of the depth maps estimated from the proposed method in comparison with those from the traditional method in some cases.

Keywords: Multi-view Video Coding, Depth Estimation Reference Software, Graph Cut.

TÓM TẮT

Với sự phát triển của công nghệ mới trong ngành công nghiệp giải trí, ti vi góc nhìn tự do, thế hệ tiếp theo của phương tiện truyền thông, sẽ cho người dùng một trải nghiệm hoàn toàn mới về ti vi khi họ có thể tự do thay đổi góc nhìn. Ti vi tương lai sẽ không chỉ hiển thị hình ảnh mà còn cho người dùng "sống" trong khung cảnh 3D. Một hướng tiếp cận đơn giản cho ti vi đa góc nhìn là sử dụng công nghệ hiện có của video đa góc nhìn với cả một hệ thống máy quay để chụp lại khung cảnh. Hình ảnh ở các góc nhìn không có camera phải được tổng hợp với sự hỗ trợ của thông tin độ sâu. Luận văn này sẽ tìm hiểu về Depth Estimation Reference Software (DERS) của Moving Pictures Expert Group (MPEG), phần mềm tham khảo để ước lượng độ sâu từ các video màu chụp bởi các máy quay đa góc nhìn. Đồng thời khóa luận cũng sẽ đưa ra phương pháp mới sử dụng lưu trữ thông tin nền để cải tiến phần mềm tham khảo. Kết quả thí nghiệm cho thấy sự cái thiện chất lượng ảnh độ sâu của phương pháp được đề xuất khi so sánh với phương pháp truyền thống trong một số trường hợp.

Từ khóa: Nén video đa góc nhìn, Phần mềm Ước lượng Độ sâu Tham khảo, Cắt trên Đồ thị

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

DERS	Depth Estimation Reference Software
VSRS	View Synthesis Reference Software
SADERS	Semi-Automatic Depth Estimation Reference Software
FTV	Free viewpoint Television
MVC	Multi-view Video Coding
3DV	3D Video
MPEG	Moving Pictures Expert Group
PSNR	Peak Signal-to-Noise Ratio
HEVC	High Efficiency Video Coding
GC	Graph Cut

Chapter 1

INTRODUCTION

1.1. Introduction and motivation

The concept of free-viewpoint Television (FTV) was first proposed by Nagoya University at MPEG conference in 2001, focusing on creating a new generation of 3D medium which allows watchers to freely change their viewpoints [1]. To achieve this goal, MPEG has been conducting a range of international standardization activities divided into two phases: Multi-view Video Coding (MVC) and 3D Video (3DV). Multi-view Video Coding, the first phase of FTV, was started in March 2004 and completed in May 2009, targeting on the coding part of FTV from the ray captures of multi-view cameras, compression and transmission of images to synthesis of new views. On the other hand, the second phase 3DV started in April 2007 was about serving these 3D views on different types of 3D displays [1].

In the basic configuration of FTV system, as shown in the Figure 1, 3D scene is fully captured by a multi-camera system. The captured images are, then, corrected to eliminate "the misalignment and luminance differences of the cameras" [1]. Then, corresponding to each corrected image, a depth map is estimated. Along with the color images, these depth maps all are compressed and transmitted to the user side. The idea of

calculating the depth maps at sender sides and sending them along with the color images helps reducing the computational work of the receiver. Moreover, it allows FTV system to be able to show the infinite number of views based on the finite number of coding views [2]. After being uncompressed, the depth maps and existing views are used to generate new views, which fully describe the original 3D scene from any viewpoints which the users want.

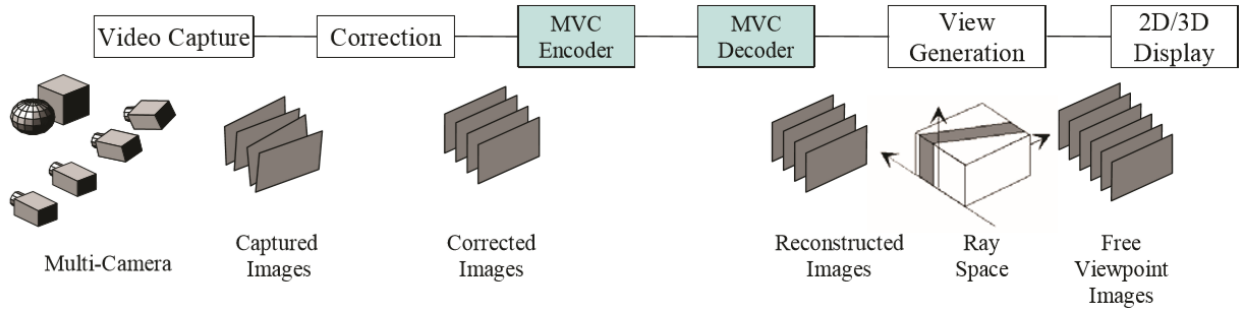


Figure 1. Basic configuration of FTV system [1].

Although depth estimation only works as an intermediate step in the whole coding process of MVC, it actually is a crucial part, since depth maps are the key idea to interpolate free viewpoints. In the sequences of MVC standardization activities, Depth Estimation Reference Software (DERS) was introduced to MPEG as a reference software for estimating depth maps from sequences of images captured by an array of multiple cameras. At first, there is only one fully automatic mode in DERS; however, as in many cases, the inefficiency of depth estimation of the automatic mode of DERS leads to the low quality of synthesized views, new semi-automatic modes were added to improve the performance of DERS and the quality of the synthesized views. These new modes, nevertheless, share a same feature which is that a very good frame having manual support but poor performance in the next ones.

1.2. Objectives

The objectives of this thesis are about understanding and learning technologies in the Depth Estimation Reference Software (DERS) of MPEG. Moreover, in this thesis, I introduce a new method to improve the performance of DERS called background

enhancement. The basic idea of this method is storing the background of the scenes and using them to estimate the separation between the foreground and the background. The color map and depth map of background are stored overtime from the first frame. Since the background does not change too much over the sequence, these maps can be used to support the depth estimation process in DERS.

1.3. Organization of the thesis

Chapter 2 is spent describing the theories, structures, techniques and modes of DERS. Among them, there is a temporal enhancement method, based on which, I developed a method to improve the performance of DERS. My method will be described clearly in Chapter 3. The setup and the results of experiments to compare the method with the original DERS is illustrated in Chapter 4 along with further discussion. The final Chapter, Chapter 5, will conclude the overall information of this thesis.

DEPTH ESTIMATION REFERENCE SOFTWARE

2.1. Overview of Depth Estimation Reference Software

In April 2008, Nagoya University for the first time has proposed the Depth Estimation Reference Software (DERS) to the 84th MPEG Conference in Archamps, France in the document [3]. In this document, Nagoya has provided all the specification and also the usage of DERS. The initial algorithm of DERS, nonetheless, had already been presented in previous MPEG documents [4] and [5]; it included three steps: a pixel matching step, a graph cut and a conversion step from disparity to depth. All of these techniques had already been used for years to estimate depth from stereo cameras. However, while a stereo camera consists of only two co-axial horizontally aligned cameras, a multi-view camera system often includes multiple cameras which are arranged as a linear or circular array. Moreover, the input of DERS is not only color images but also a sequence of images or a video, which requires a synchronization for the capture time of cameras in the system. The output of DERS, therefore, is also a sequence which each frame is a depth map corresponding to a frame of color sequences. Since the first version, many improvements have been made in order to enhance the quality of depth maps: Sub-pixel precision at DER1.1, temporal consistency at DERS 2.0, Block Matching and Plane Fitting at DER 3.0… However, because of the inefficiency of traditional automatic DERS, in DERS 4.0 and 4.9, semi-automatic modes and then reference mode have been respectively introduced as alternative approaches. In semi-automatic DERS (or SADERS), manual

input files are provided at some specific frames. With the power of temporal enhancement techniques, the manual information is propagated to next frames to support the depth estimation process. On the other hand, reference mode takes an existing depth sequence from another camera as a reference when it estimates a depth map for new views. Until the latest version of DERS, new techniques have been kept integrating into it to improve the performance. In July 2014, DERS software manual for DERS 6.1 has been released [6].

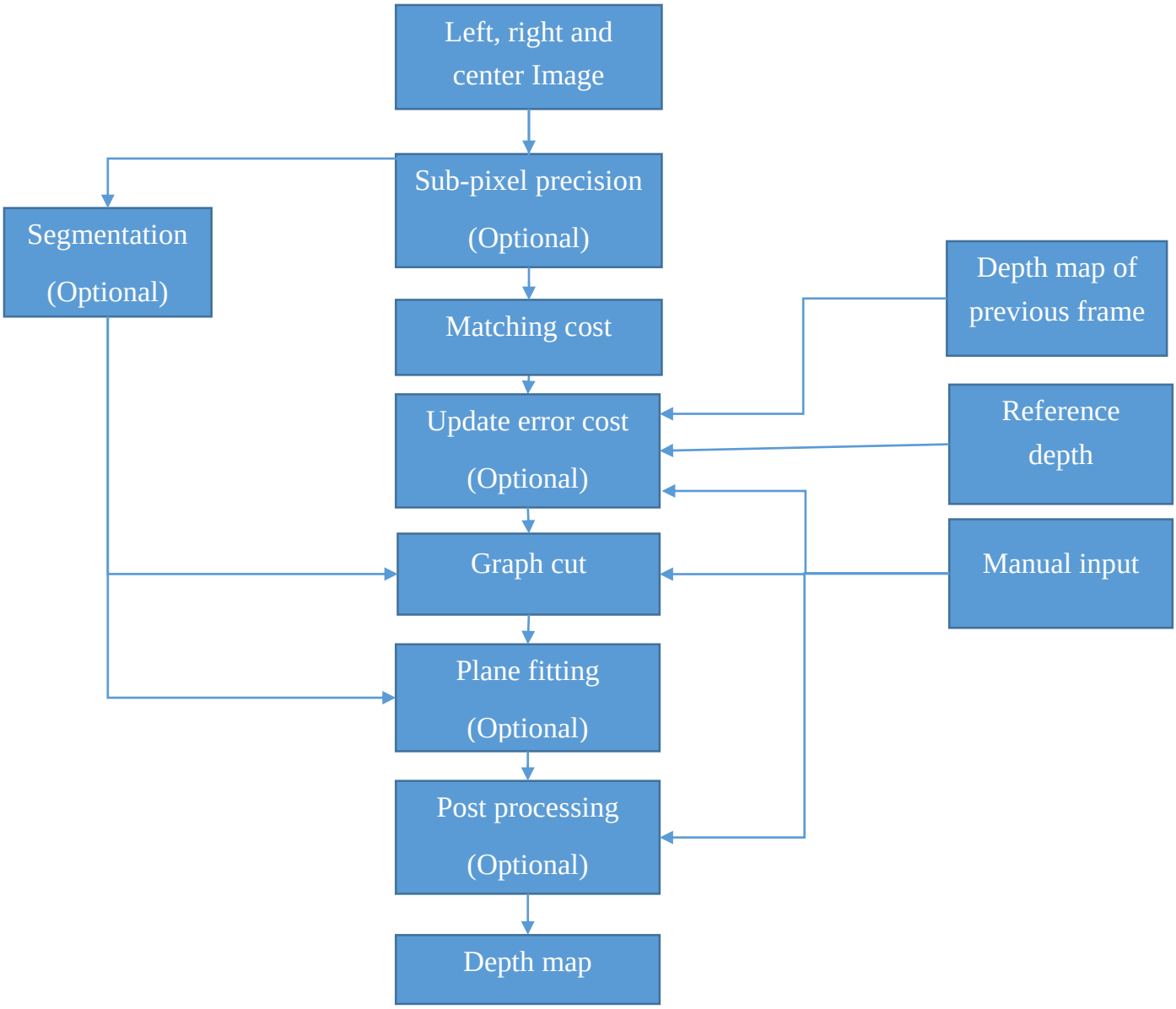


Figure 2. Modules of DERS

After six versions of DERS have been released, the configuration of DERS has become more and more intricate with various techniques and methods. Figure 2 shows the modules and the process of depth estimation of DERS.

As it can be seen from Figure 2, while most of modules are optional, there are still two modules (matching cost and graph cut) that cannot be replaceable. As mentioned above, these two modules have existed from the initial version of DERS as the key for estimating depth. The process of estimating depth starts at each frame in the sequence with three images: left, center and right images. The center image is actually the frame at the center camera view and also the image we want to calculate the corresponding depth map. In order to do so, it is required to have a left image from the camera in the left of the center camera and a right image from the camera in the right of the center camera. It is also required that these images are synchronized in the capture time. These images are, then, passed to an optional sub-pixel precision module, which us interpolation methods to double or quadruple the size of the left and right images to increase the precision of depth estimation. The matching cost module, as its name, finds a value to match the pixel of the center image with those of left or right images. Although there are several methods to calculate the matching cost, values from these share a same property that the smaller they are, the higher chance two pixels are matched. These matching values are then modified as some additional information is added to them before it goes to the graph cut module. A global energy optimization technique, graph cut, is used to label each pixel to a suitable depth or disparity based on the matching cost values, additional information and the smoothness property. Segmentation can also be used to support the graph cut optimization process as it divides the center image into segments, pixels in each of which are likely to have the same depth. After the graph cut process, a depth map has already been generated; however, for better depth quality, the plane fitting and post processing steps can be optionally used. While the plane fitting method smoothens depth values of pixels in a segment by considering it as a plane in space, the post processing, which appears only in the semi-automatic modes, reapplies the manual information into the depth map.

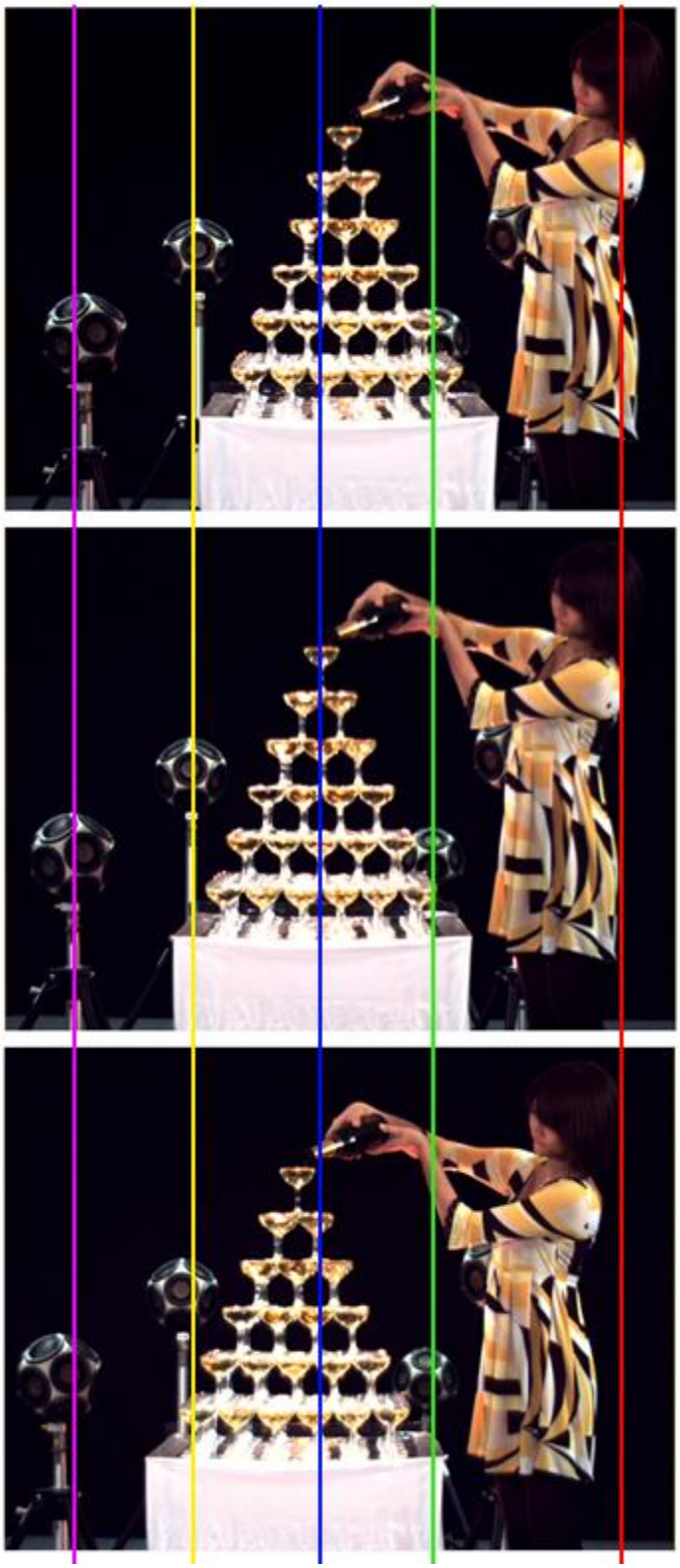


Figure 3. Examples of the relation between disparity and depth of objects

2.2. Disparity - Depth Relation

All algorithms to estimate depth for multi-view coding or even for stereo camera are all based on the relation between depth and *disparity*. "The term *disparity* can be looked upon as horizontal distance between two matching pixels" [7]. The Figure 3**Error! eference source not found.** can illustrate this relation. The three images in Figure 3 from top to bottom are taken respectively from Camera 37, 39 and 41 of Sequence Champagne of Nagoya University [8]. It can be seen that objects, which are further to the camera system, tend to move horizontally to the left less than the nearer ones. While the girl and the table, which is near the capture plane, moves over views, the furthest speaker nearly stays at its position in both three images. This phenomenon can be explained by camera pinhole model and mathematics with the Figure 4.

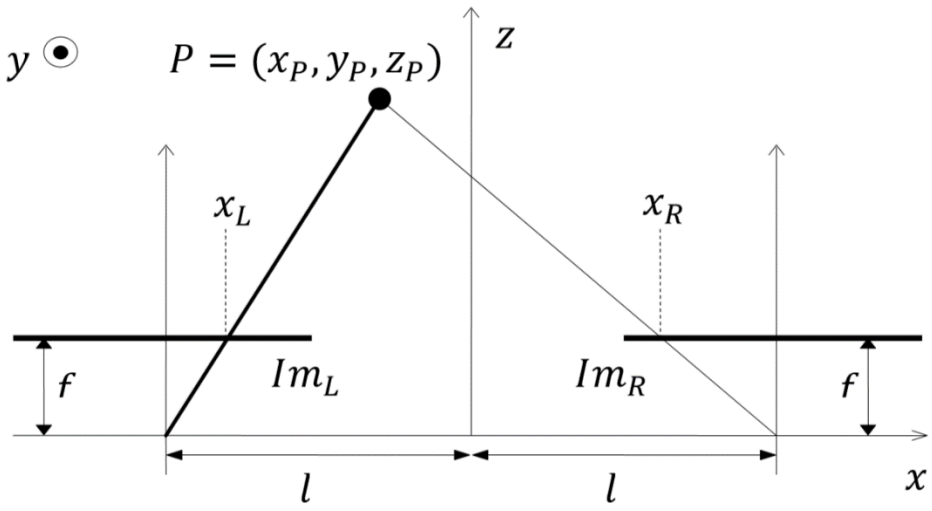


Figure 4. The disparity is given by the difference $d = x_L - x_R$, where x_L is the x-coordinate of the projected 3D coordinate x_P onto the left camera image plane Im_L and x_R is the x-coordinate of the projection onto the right image plane Im_R [7].

From the Figure 4, [7] has proved that the distance of images of an object (or disparity) is inversely proportional to the depth of that object:

$$d = x_L - x_R = f\left(\frac{x_P + l}{z_P} - \frac{x_P - l}{z_P}\right) = \frac{2fl}{z_P} \quad (1)$$

where

d is the disparity or the distance of images of object-point P captured by two cameras,

x_L, x_R are the coordinates of images of object-point P

f is the focal length of both cameras,

$2l$ is the distance between two cameras,

z_P is the depth of the object-point P.

As it was proved that the depth and the disparity of an object is inversely proportional, the problem of estimating the depth turned into that of calculating the disparity or finding a matching pixel for each pixels in the center image.

2.3. Matching cost

To calculate the disparity of each pixel in the center image, it is required to match those pixels with their correspondences in the left and the right images. As mentioned before, input images of DERS are all corrected to eliminate difference of illumination and synchronized in capture time. We, therefore, can assume that intensities of matching pixels of same object-points are almost similar. This assumption is also the key to estimate matching pixels.

To reduce the complexity of computation, cameras are aligned horizontally. Moreover, the image sequences are all rectified, which makes the matching pixels align in a same horizontal level. In other words, instead of looking all over the left or right images for a single matching pixel, we only need to find it in one horizontal row.

Using two mentioned above ideas, matching cost or error cost functions are formed to help find the matching pixels. They all share the property that the smaller value the function responds the higher chance it is the matching pixel we are looking for.

2.3.1. Pixel matching

The pixel matching cost function is the simplest matching cost function in DERS. It appeared in DERS from the initial version introduced by Nagoya University in [4]. For each pixel in the center image and each disparity in a predefined range, DERS evaluates matching cost function by calculating the absolute intensity difference between the pixel in the center image and those in the left and right images respectively and choosing the minimum value. Therefore, the smaller result is that the more similar intensities of pixels and the more likely those pixels are matching. For more specific, we have the below formula:

$$C(x, y, d) = \min\big(C_L(x, y, d), C_R(x, y, d)\big), \tag{2}$$

where

$$C_L(x, y, d) = |I_C(x, y) - I_L(x + d, y)|$$

$$C_R(x, y, d) = |I_C(x, y) - I_R(x - d, y)|$$

2.3.2. Block matching

To improve the performance of DERS, the document [9] presented a new matching method called block matching. While a pixel matching cost function compares pixel to pixel, the block matching cost function works with window comparison. For more specific, when matching two pixels with each other, the block matching method concerns about comparing windows containing those pixels. The main advantage of this method over the pixel matching method is that it reduces noise sensitivity. However, this advantage comes along with a disadvantage, which is loss of detail and more computation when a bigger window size is selected [7]. DERS, therefore, only uses 3x3 windows with matching pixels at their center:

$$C(x,y,d) = \min\big(C_L(x,y,d), C_R(x,y,d)\big), \quad (3)$$

Where

$$C_L(x,y,d) = \frac{1}{9}\sum_{i=x-1}^{x+1}\sum_{j=y-1}^{y+1} |I_c(i,j) - I_L(i+d,j)|$$

$$C_R(x,y,d) = \frac{1}{9}\sum_{i=x-1}^{x+1}\sum_{j=y-1}^{y+1} |I_c(i,j) - I_R(i-d,j)|$$

For pixels at the corners or edges of images, where the 3x3 windows do not exist, pixel matching or smaller block matching (2x2, 2x3 or 3x2) are used.

2.3.3. Soft-segmentation matching

Similar to the block matching, soft-segmentation matching method also uses aggregation windows in comparison [10]. However, each pixel in the block window is weighted differently by its distance and intensity similarity to the center pixel; this feature resembles to the bilateral filtering technique [7]. Moreover, the size of window of soft-segmentation in DERS can be changed in the configuration file and it is normally quite large as the default value is 24x24. Soft-segmentation matching, therefore, takes much more time for computing than block matching and pixel matching. Below is the formula of soft-segmentation matching cost function:

$$C(x,y,d) = \min\big(C_L(x,y,d), C_R(x,y,d)\big), \quad (4)$$

where

$$C_L(x,y,d) = \frac{\sum_{(i,j)\epsilon\, w(x,y)} W_L(i,j,x,y) W_C(i+d,j,x+d,y) |I_C(i,j) - I_L(i+d,j)|}{\sum_{(i,j)\epsilon\, w(x,y)} W_L(i,j,x,y) W_C(i+d,j,x+d,y)}$$

$$C_R(x,y,d) = \frac{\sum_{(i,j)\,\epsilon\, w(x,y)} W_R(i,j,x,y) W_C(i-d,j,x-d,y) |I_C(i,j) - I_R(i-d,j)|}{\sum_{(i,j)\,\epsilon\, w(x,y)} W_R(i,j,x,y) W_C(i-d,j,x-d,y)}$$

and

$w(x,y)$ is a soft-segmentation window center at (x,y)

$W(i,j,x,y)$ is the weight function for the pixel (i,j) in the window centered at (x,y):

$$W(i,j,x,y) = e^{-\frac{|I(x,y)-I(i,j)|}{\gamma_C} - \frac{|(x,y)-(i,j)|}{\gamma_d}}$$

2.3.4. Epipolar Search matching

As mentioned above, all images are rectified to reduce the complexity in searching for matching pixels since we only have to make a search in a horizontal line instead of the whole image. However in document [11], authors from Poznan University of Technology pointed out that "in the case of sparse or circular camera arrangement", rectification "distort the image at unacceptable level" as in Figure 5**Error! Reference source not found.**.

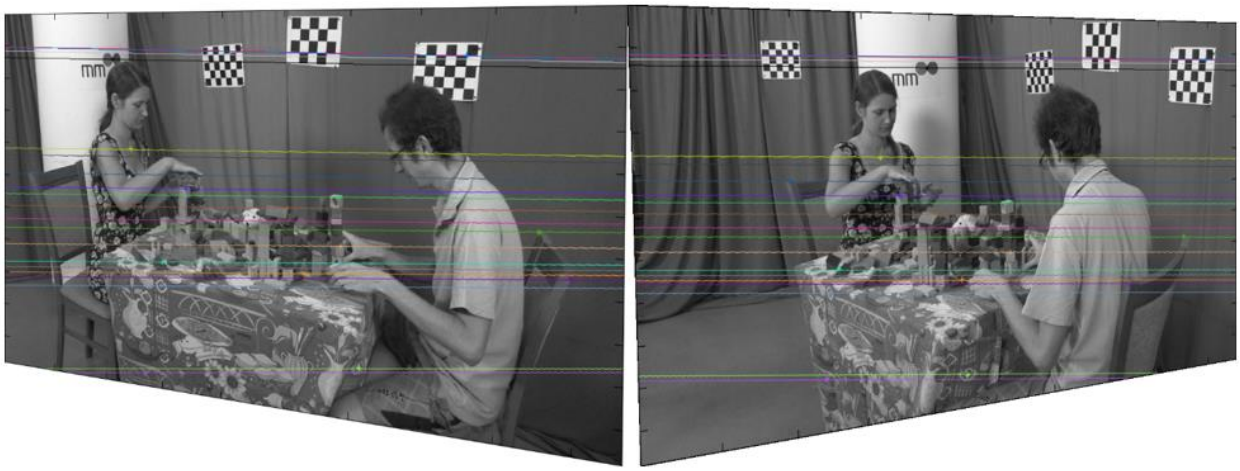


Figure 5. Exampled rectified pair of images from "Poznan_Game" sequence [11].

They, therefore, suggested that instead of applying rectification to images before matching, DERS should do all kinds of matching methods (pixel, block or soft-segmentation) along epipolar lines which can be calculated based on camera parameters [11] like in Figure 6.

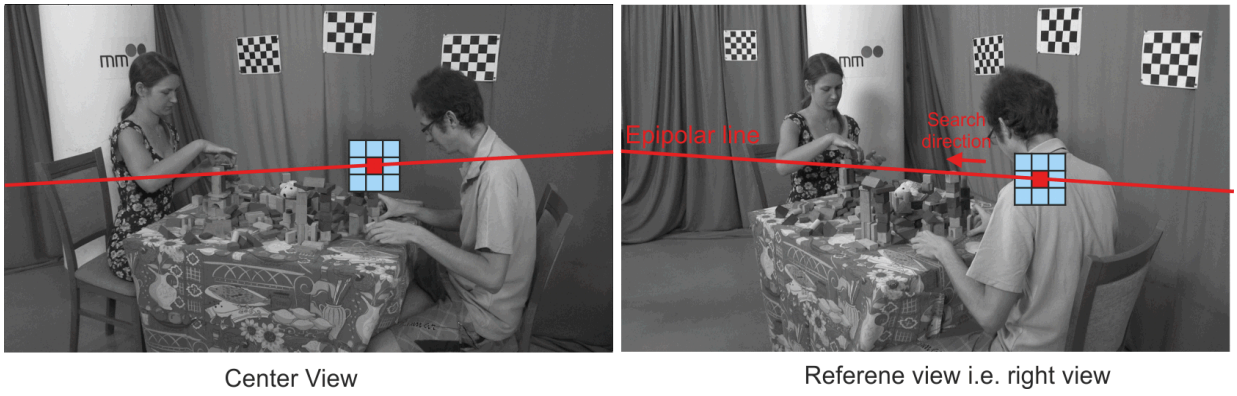


Figure 6. Explanation of epipolar line search [11].

2.4. Sub-pixel Precision

Normally, a depth map is a grayscale image, whose pixels have values in range from 0 to 255. However, the disparity value is only an integer staying in a range from 0 to no more than 100, which makes the disparity value and the depth value do not map injectively. In other words, in some cases, the integer disparity value does not match with the requirements of the depth value. That is why sub-pixel technique was brought to DERS in document [12]. The idea of sub-pixel technique is that estimating the disparity value accurately at sub-pixel precision by interpolating the left and right images on sub-pixel positions using bi-linear or bi-cubic filter (Figure 7). So that the half-pixel doubles the number of possible disparity values while the quarter-pixel quadruples it. Although sub-pixel precision approach create a more accurate depth map, it required more computation as the size of the left and right images are multiplied (Figure 8). Moreover, in the case of epipolar line search matching method, since the search runs along epipolar line not only the horizontal row, not only the width but also the height of the left and right images are interpolated to double or quadruple of their sizes.

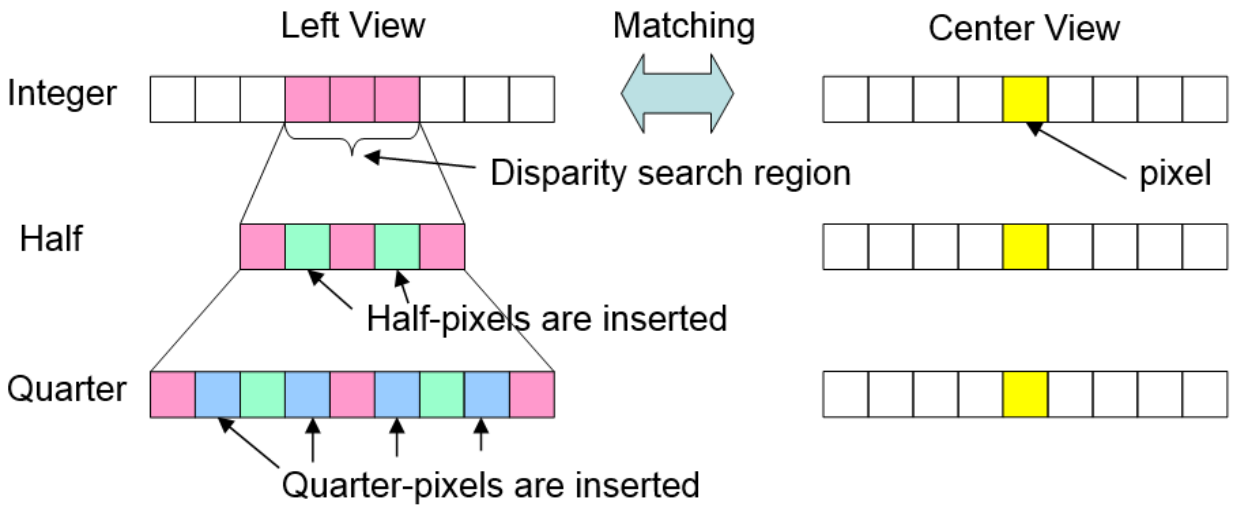


Figure 7. Matching precisions with searching in horizontal direction only [12]

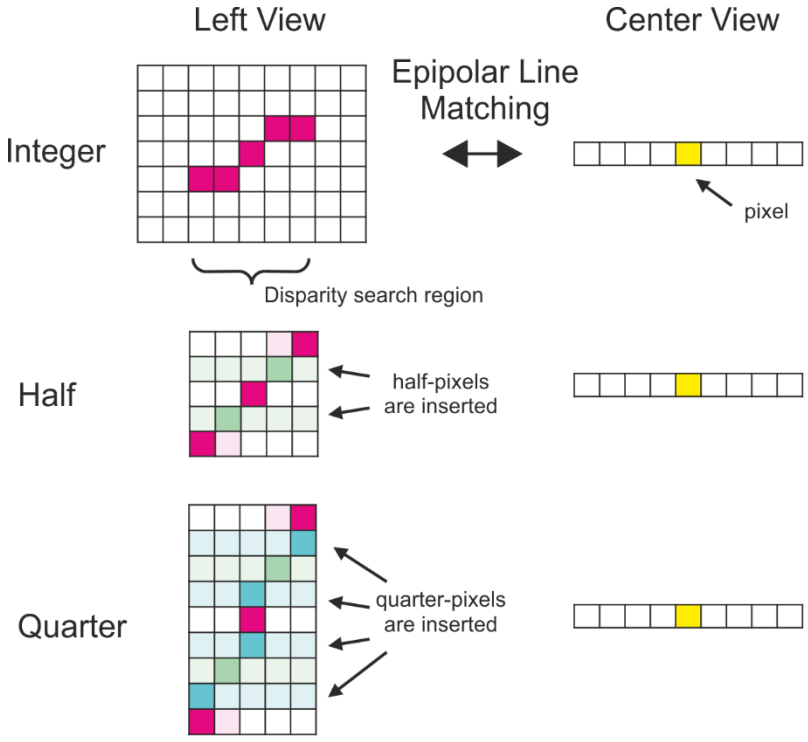


Figure 8. Explanation of vertical up-sampling [11].

Although sub-pixel technique provides a more accurate depth map, in document [13], two authors Olgierd Stankiewicz and Krzysztof Wegner from Poznań University of

Technology have made experiments to prove that "Pixel-precision mode in Depth Estimation Reference software has no impact on quality of synthesized views" (the final target of MVC). However, this mode is still kept as a part of DERS.

2.5. Segmentation

Unlike other process which affects directly to the center images in its path of transforming to depth images, segmentation is only a support step. For more specific, segmentation is applied to the center image to divide the image into segments, pixels of which have similar color values; this result, then, is provided to Graph Cut or Plane Fitting modules to create and improve the depth map. Segmentation technique is used because of the likelihood of similar intensities of pixels in a same object. In DERS, segmentation is implemented by using one in three segmentation functions of OpenCV: cvPyrMeanShiftFiltering, cvPyrSegmentation and cvKMeans2. Figure 9 is an example of segmentation:

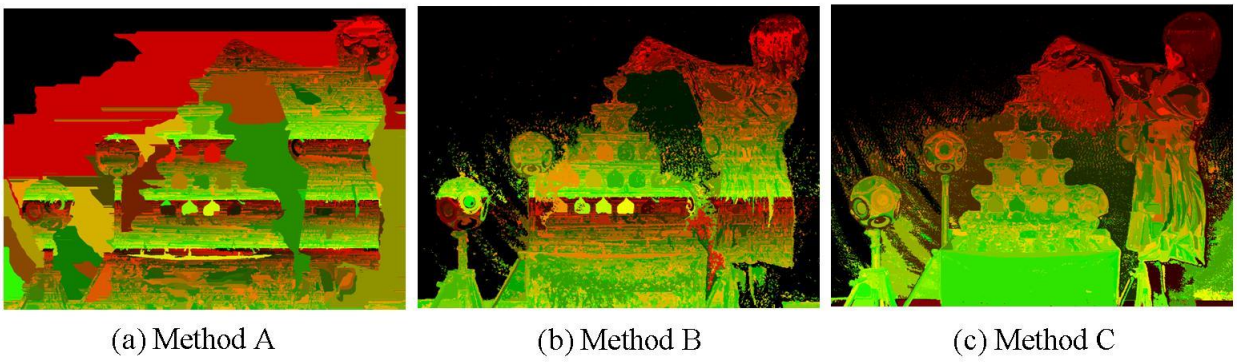


Figure 9. Color reassignment after Segmentation for invisibility. From (a) to (c): cvPyrMeanShiftFiltering, cvPyrSegmentation and cvKMeans2 [9].

However, the idea of using segmentation to support depth estimation also comes with a negative influence since the color segments are not always consistent with the depth areas.

2.6. Graph Cut

Existed in DERS from the very initial version, Graph Cut is unarguably the most important step in the whole process of depth estimation of DERS. Graph Cut is a global optimization method which estimates disparities of all pixels of the center image at once by minimizing a single energy function formed by matching costs, relations between pixels and other additional information [7].

2.6.1. Energy Function

In stereo research, there are two groups of methods to tackle the problem of assigning disparity values to each pixel. One of them is the group of local methods which estimates the disparity values only based on information in a neighborhood area around the pixel. On the other hand, the other group of global methods, which Graph Cut is one of them, tries to determine the disparity values of all pixels at once by using all information. All pixels are linked to each other in a single energy function that when it is optimized, it gives us the disparities of all pixels. This energy function consists of two terms: data term and smooth term:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (5)$$

where

d is the disparity map and $d(x, y)$ is the disparity at pixel (x, y) and

λ is the configurable coefficient to balance values of data term and smooth term.

The data term measures the dissimilarity of d with existing information, while the smooth term concerns about the discontinuity of values between adjacent pixels in the disparity map. In the simplest version Graph Cut, the data term is a total aggregation of pixel matching costs at each pixel position and the smooth term is the absolute differences of adjacent pixels:

$$E_{data}(d) = \sum_{(x,y) \in I_C} C\big(x, y, d(x,y)\big) \tag{6}$$

$$E_{smooth}(d) = \sum_{(i,j)} \sum_{(x,y)\ adjacent\ (i,j)} V(x, y, i, j, d(x,y), d(i,j)) \tag{7}$$

where

$$V(x, y, i, j, d, d') = |d - d'|$$

The smooth term in the newer version is improved with the help of segmentation. As mentioned above, the segmentation based on pixel color can be used as support information for disparity estimation since the pixels on a same object likely have similar color:

$$E_{smooth}(d) = \sum_{(i,j)} \sum_{(x,y)\ adjacent\ (i,j)} V(x, y, i, j, d(x,y), d(i,j)) \tag{8}$$

where

$$V(x, y, i, j, d, d') = \beta |d - d'|$$

$$\begin{cases} \beta = 1 \ if\ segment(x,y) = segment(i,j) \\ 0 < \beta < 1 \ if\ segment(x,y) \neq segment(i,j) \end{cases}$$

With the improvement of matching cost functions, the data term also changes from the sum of pixel matching costs to that of block matching costs or that of soft-segmentation matching costs. Moreover, as MVC uses a multi-view camera system to record a 3D scene in sequences of images, the correlation between frames and frames, between images from different cameras can be used to have better estimation. The information from correlation between frames in a sequence, or temporal information, will be discussed in the Temporal Consistency Section of this Chapter. The usage of the information from different cameras is in the reference mode of DERS. However, I will not discuss more about it in this thesis.

2.6.2. Optimization

Once the energy function has been built, minimizing values of this function will give us the small dissimilarity of disparity values with observed data (matching costs) and with other disparity values, which basically shows a good depth map. In order to optimize energy function, in [14], Graph Cut solution based on maximum flow/ minimum cut algorithm has been introduced. Although two approaches $\alpha - \beta\ swap$ and $\alpha - expansion$ were presented, only $\alpha\ - expansion$ was used in DERS.

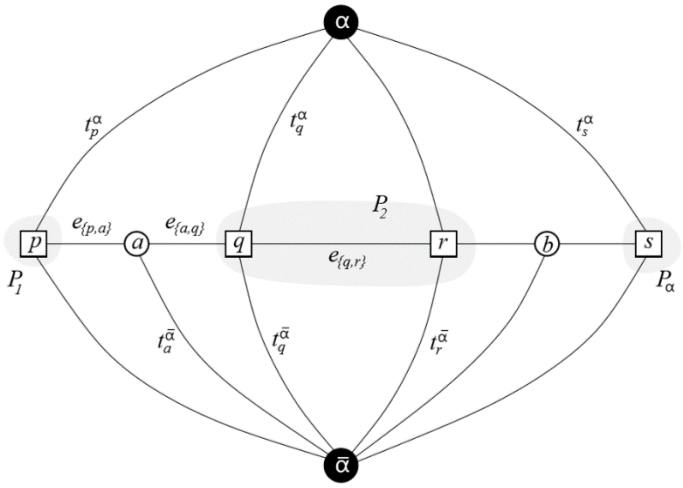


Figure 10. An example of G_α for a 1D image. The set of pixels in the image is $V = \{p, q, r, s\}$ and the current partition is $P = \{P_1, P_2, P_\alpha\}$ where $P_1 = \{p\}$, $P_2 = \{q, r\}$, and $P_\alpha = \{s\}$. Two auxiliary nodes $a = a_{\{p,q\}}$, $b = a_{\{r,s\}}$ are introduced between neighboring pixels separated in the current partition. Auxiliary nodes are added at the boundary of sets P_l [14].

At first, every pixels in the disparity map is set as 0. In each loop, every possible disparity values in our predefined range are used to apply $\alpha - expansion$. To optimize energy function, in each application of $\alpha - expansion$, the whole center image is turned into a graph which has pixels as its vertices. Two more nodes, a source and a sink, are added into the graph and are connected to other nodes. The source is the α node while the sink is $\bar{\alpha}$. For each pair of adjacent pixels, if two pixels have already shared the same disparity value, there will exist an edge between two corresponding vertices in the graph;

if not, a new additional node is put in and linked to both of them and also to the sink. The Figure 10 illustrated this idea of Graph Cut. The edges of the graph are weighted differently based on the current labels of its terminals, which is shown in Table 1.

Table 1. Weights assigned to edges in Graph Cut.

Edge	Weight	For
$t_p^{\bar{\alpha}}$	∞	$p(x,y) \in P_\alpha$
$t_p^{\bar{\alpha}}$	$C(x,y,d(x,y))$	$p(x,y) \notin P_\alpha$
t_p^{α}	$C(x,y,\alpha)$	$p(x,y) \in P$
$e_{\{p,a\}}$	$V(x,y,i,j,d(x,y),\alpha)$	$p(x,y)\ adjacent\ q(i,j)$ $d(x,y) \neq d(i,j)$
$e_{\{a,q\}}$	$V(x,y,i,j,\alpha,d(i,j))$	
$t_a^{\bar{\alpha}}$	$V(x,y,i,j,d(x,y),d(i,j))$	
$e_{\{p,q\}}$	$V(x,y,i,j,d(x,y),\alpha)$	$p(x,y)\ adjacent\ q(i,j)$ $d(x,y) = d(i,j)$

With the source, the sink and all the weights of edges, we can apply the minimum cut/maximum flow algorithm to find a cut through the graph. Figure 11, a smaller version of the graph after different kinds of cut are applied, has shown us a different way to divide two nodes into two group of α disparity or not. As the minimum cut/maximum flow algorithm gives us optimal cut, it is guaranteed that each $\alpha - expansion$ turns makes the energy function smaller. However, as it required a lot of computation work, normally only two loops are run.

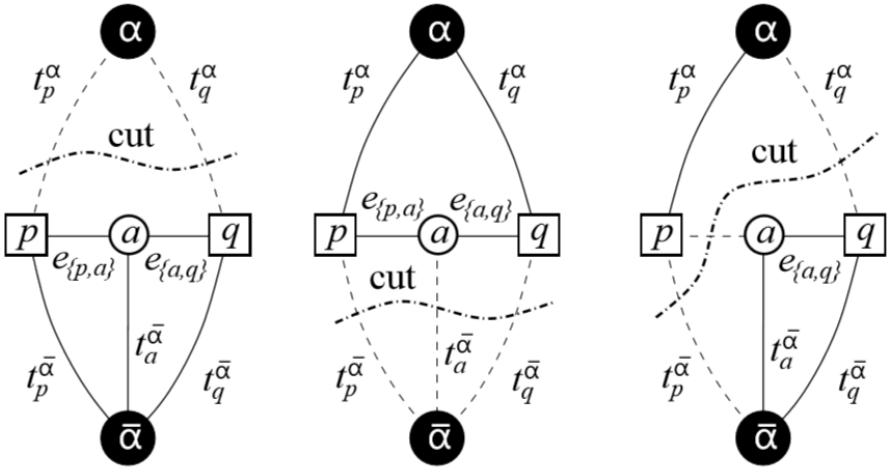


Figure 11. Properties of a minimum cut C on G_α for two pixel p,q such that $d_p \neq d_q$. Dotted lines show the edges cut by $C$and solid lines show the edges in the induced graph $G(C) = \langle V, E - C \rangle$ [14].

2.6.3. Temporal Consistency

In order to use the temporal information, some methods have been introduced to DERS, for example [15] and [16]. Nevertheless, only [16] has been kept in DERS 6.1. The depth map and color center image of the previous frame are kept as references in estimating the current frame. In Update Error module, when consistency of temporal information is required, a block motion size 16x16 is used to compare two color center images of adjacent frames. The motion search algorithm is applied only to target finding unchanged areas, which are called "background" in [16]. If the sum of absolute differences of corresponding pixel intensities between frames of all pixels in a 16x16 block is smaller than a specific threshold, the area of the block is considered as containing no motion. For all pixels in these unchanged areas, a new term is added into the data term of the energy function as:

$$E_{data}(d) = \sum_{(x,y) \in I_C} C\big(x, y, d(x,y)\big) + C_{temporal}(x, y, d(x,y)), \tag{9}$$

where

$$C_{temporal}(x, y, d(x, y)) = \alpha|d(x, y) - d_{prev}(x, y)|$$

And

$d_{prev}(x, y)$ is the disparity value of pixel (x, y) in the previous frame.

$$\alpha = \begin{cases} 1 \; if \sum_{(i,j) \in w(x,y)} |I_c(i,j) - I_{cprev}(i,j)| < Thres_{motion} \\ 0 \; otherwise \end{cases}$$

This new term adding a dissimilarity value into the overall energy function if there exists an inconsistency in the disparity values between frames. Graph cut itself will optimize the energy function to add this into the final disparity results.

2.6.4. Results

After Graph Cut, the disparity map is directly converted to the depth map by using the relation between depth and disparity. Figure 12 are examples of depth maps after Graph Cut.

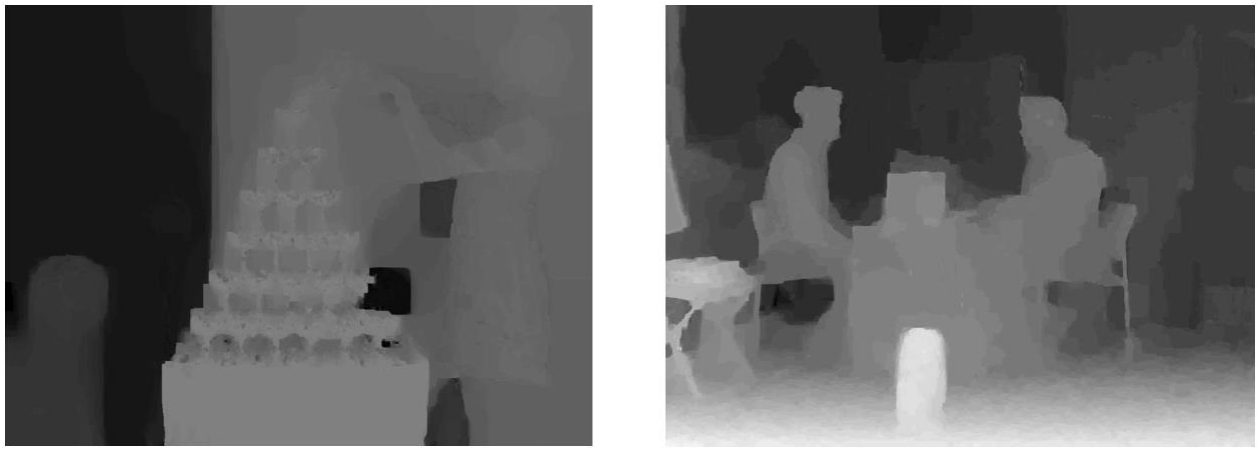


Figure 12. Depth maps after graph cut: Champagne and BookArrival [9].

2.7. Plane Fitting

Although the support information from segmentation has already been used in Graph Cut, Plane Fitting also uses this information again to improve the depth map quality. Plane fitting basically uses the idea that it considers each segments from the segmentation as a plane in the space and it fits the current depth map from Graph Cut module to models of space planes. Each plane in space is modeled with a formula:

$$d(x, y) = ax + by + c \tag{10}$$

To calculate the coefficients (a, b, c) of the above formula, all the existed pixels of the segments are considered as a point in space at (x, y, d). A least square fitting is used to fit the point into a map and calculate the coefficients:

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} y_i^2 & \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} y_i & \sum_{i=1}^{m} 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i d_i \\ \sum_{i=1}^{m} y_i d_i \\ \sum_{i=1}^{m} d_i \end{bmatrix} [9] \tag{11}$$

After having the coefficients (a, b, c) and the equation of the segment plane in space, the depth of each pixel-point is recalculated so that it really stays inside the plane.

The result of the plane fitting is shown in Figure 13.

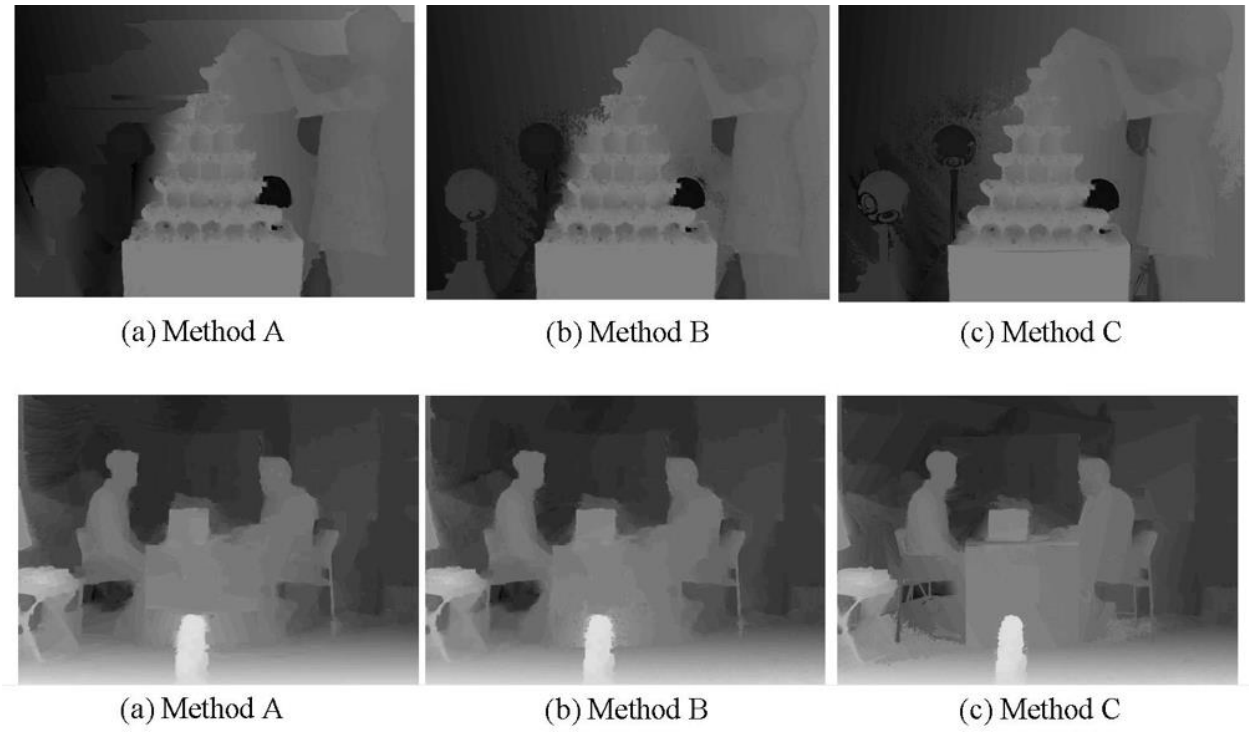


Figure 13. Depth maps after Plane Fitting. Left to Right:: cvPyrMeanShiftFiltering, cvPyrSegmentation and cvKMeans2. Top to bottom: Champagne, BookArrival [9].

2.8. Semi-automatic modes

Because of the inefficiency of the automatic mode of DERS, in [17], semi-automatic DERS (SADERS) has been proposed. The objective of SADERS is to use additional manual information to improve the accuracy of the depth map result. Until now, there are three different modes of SADERS which has been integrated into DERS.

2.8.1. First mode

The first mode of SADERS is introduced as SADERS 1.0 in [17]. This mode targets in using the temporal consistency technique to propagate information from manual depths which can be created in some frames. Manual depth maps are provided at some frame positions. When DERS goes the depth map at these positions, it uses the manual depth

maps as results. In different frame positions, DERS uses these manual depth maps or previous estimated depth maps as references in the temporal consistency mode of Graph Cut. A flowchart of SADERS 1.0 is shown in Figure 14.

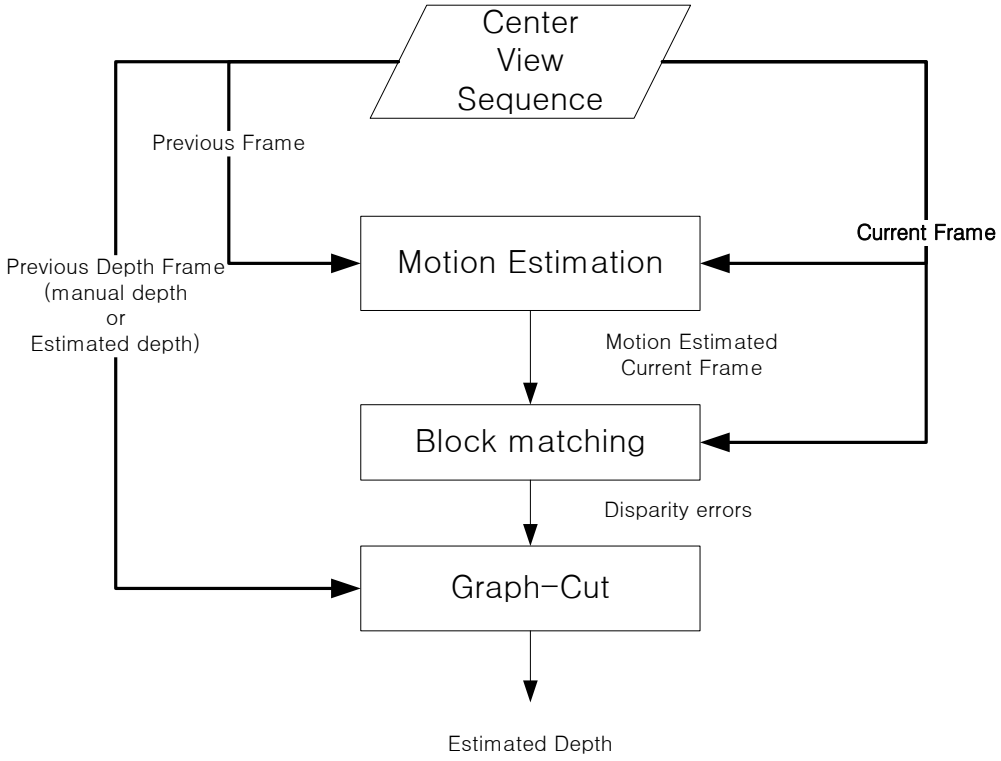


Figure 14. Flow chart of the SADERS 1.0 algorithm [17].

2.8.2. Second mode

The second mode of SADERS is provided in [18]. The new version of SADERS still uses the temporal consistency property to propagate manual information. However, instead of providing a whole depth map for DERS, it provides two different kinds of manual information (a manual disparity map and a manual edge map) which helps DERS itself make a more accurate estimation; it also provides a manual static map to enhance the temporal consistency (Figure 15).

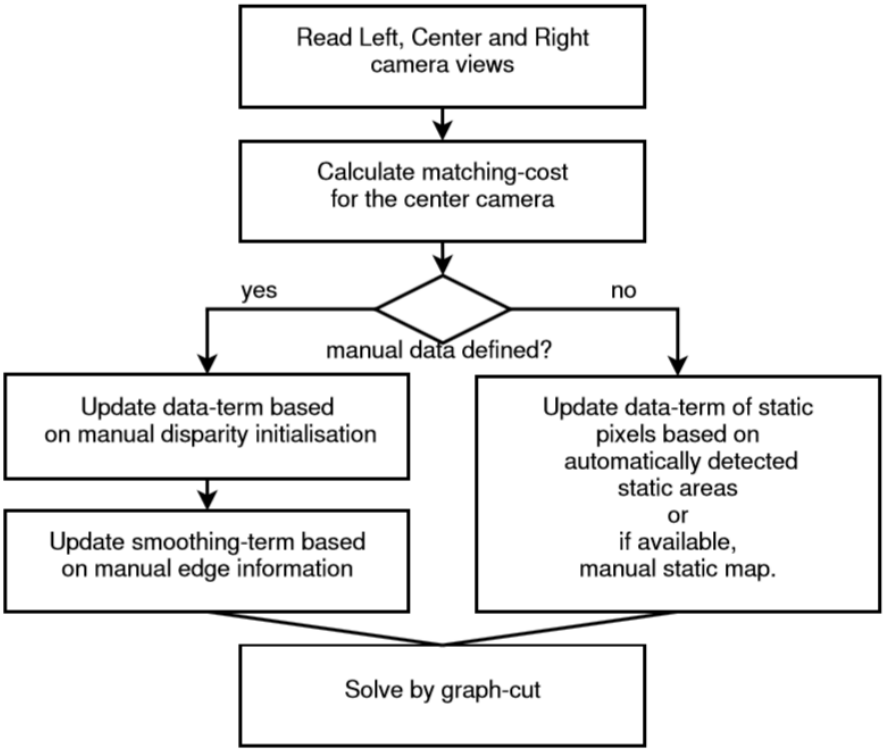


Figure 15. Simplified flow diargram of the second mode of SADERS [18].

The manual disparity map contains areas with given disparity; however, not all the pixels of the center image are provided with their disparities but only "where automatic depth estimation fails to find an accurate value" [18]. This manual disparity map is used as the initialization for the data term of Graph Cut. The manual static map, on the other hand, shows areas whose depths are not changed over time so that the depths estimated by manual information are preserved.

In initial stage:

$$E_{data}(d) = \begin{cases} C(x,y,d(x,y)) \;\; if \;\; MD(x,y) = 0 \\ 0 \;\; if \;\; MD(x,y) = d(x,y) \\ 2C(x,y,d(x,y)) \end{cases} \quad (12)$$

In temporal stage:

$$E_{data}(d) = \begin{cases} 0 & if\ MS(x,y) = static\ and\ d(x,y) = d_{init}(x,y) \\ 2C\big(x,y,d(x,y)\big) & if\ MS(x,y) = static\ \ and\ d(x,y) \neq d_{init}(x,y) \\ C\big(x,y,d(x,y)\big) + C_{temporal}\big(x,y,d(x,y)\big) & if\ temporal\ consistency \\ C\big(x,y,d(x,y)\big) & otherwise \end{cases} \quad (13)$$

where

temporal consistency:$\sum_{(i,j)\,\in\, w(x,y)} \big|I_c(i,j) - I_{cprev}(i,j)\big| < Thres_{motion}$ like (9)

The manual edge map is a binary map which is used to support the smooth term of the energy function since it provides the separation between different depth areas. A new scaling factor β is added into the smooth term wherever the manual edge map indicates an edge.

$$E_{smooth}(d) = \sum_{(i,j)} \sum_{(x,y)\ adjacent\ (i,j)} V(x,y,i,j,d(x,y),d(i,j)) \quad (14)$$

where

$$V(x,y,i,j,d,d') = \beta|d - d'|$$

$$\begin{cases} \beta = 1\ if\ ME(x,y) = 0\ and\ ME(i,j) = 0 \\ 0 < \beta < 1\ if\ ME(x,y) = 1\ or\ ME(i,j) = 1 \end{cases}$$

In comparison with the first mode of SADERS, this mode takes less time in preparing manual information.

Below are some results from the second mode:

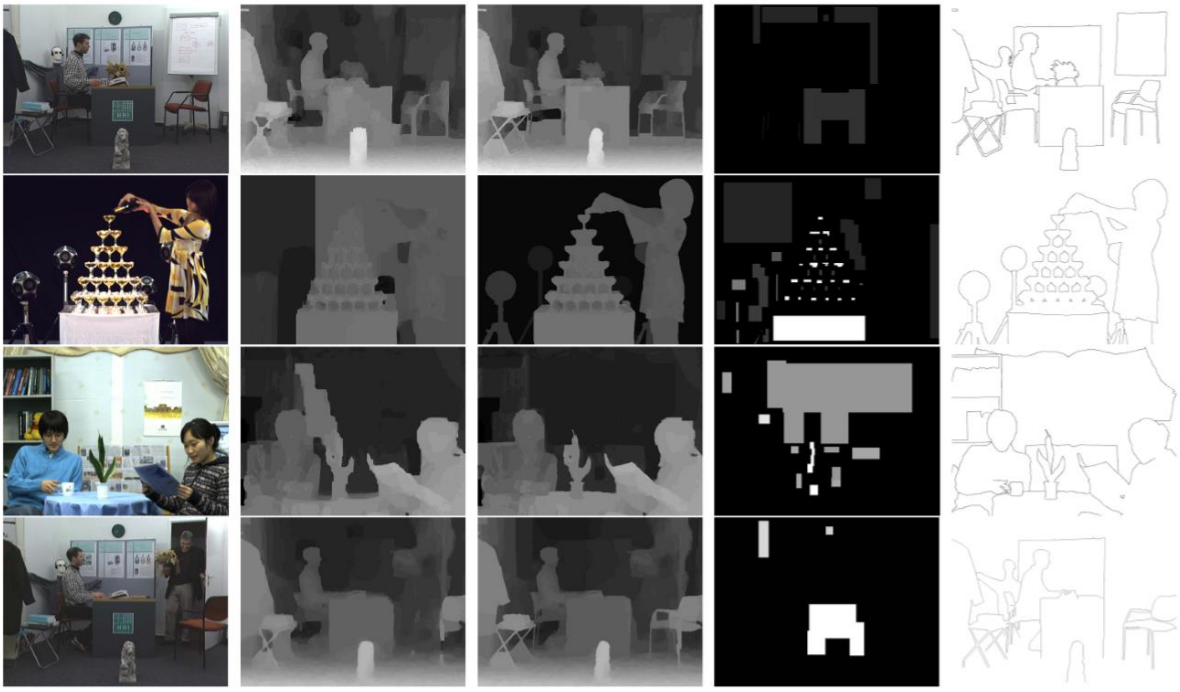


Figure 16. Left to right: camera view, automatic depth result, semi-automatic depth result, manual disparity map, manual edge map. Top to bottom: BookArrival, Champagne, Newspaper, Doorflowers and BookArrival [18].

2.8.3. Third mode

The third mode of SADERS is very same with the second one; it, however, preserved completely static areas of the manual static map and the unchanged areas detected by the temporal consistency technique by copying its depth value to next frames instead of using Graph Cut.

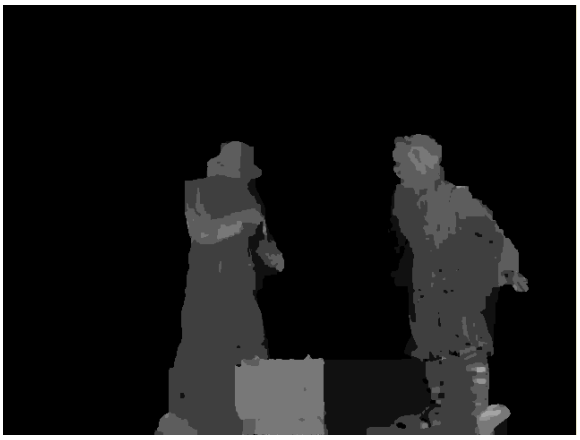
THE METHOD: BACKGROUND ENHANCEMENT

3.1. Motivation example

Although there are many modules and modes of DERS which is built to improve the performance of depth estimation process, DERS still shows the poor quality in depth estimation for low-textured area. The sequence Pantomime from [8] is an example for this sequence type with low-textured background. As can be seen from Figure 17, most of the background of Pantomime sequence is covered by dark black color. The low-textured area is difficult to estimate the depth because the matching costs (pixel matching cost, block matching cost or soft-segmentation matching cost) of pixels in this area are close to each other when the disparity value parameter changes. The pixels of the low-textured area, therefore, are easily affected by other textured pixels because of the smooth term of the energy function. For example, in SADERS, the first depth map is estimated with the help of manual information, which makes the depth of low-textured area quite accurate (Figure 19.a); however, pixels near the textured area in next frame are rapidly influenced by the depth of their textured neighbors in next frames in Figure 18.b,c,d.

Although SADERS works great in the first frame, it is unable to accurately separate the low-textured background with the textured foreground in the next frames. These examples of Pantomime motivate the method to improve performance of the DERS

Figure 17. Motivation example



a) Frame 0

b) Frame 10

c) Frame 123

d) Frame 219

Figure 18. Frames of Depth sequence of Pantomime. Figure a and b have been processed for better visual effect.

3.2. Details of Background Enhancement

The method which is called as Background Enhancement targets in improving the performance of DERS in the low-textured background situation in Pantomime sequences. Although with the help of manual information, DERS in semi-automatic mode has estimated a high quality depth map at the positions of manual frames, it fails to keep this success in the next frames (Figure 18). There are two reasons for this phenomenon. Firstly, because the low-textured background has low differences between matching costs of different disparity values, their smooth terms dominate their data terms in Graph Cut process, which makes their estimated depth results easily affected by those of textured pixels. Secondly, while the temporal consistency is the key to conserve the correct disparity value of the previous frame, it fails when detecting some non-motion background area as motion areas.

The Figure 19 shows the result of the motion search used by temporal consistency techniques. White area illustrated the area without any motion, while the rest shows the motion-detected area. As it can be seen that there are back pixels around the clowns, which basically is the low-textured no-motion area. As motions are wrongly detected in these pixels, temporal consistency term (Section 2.6.3) is not added to their data term. Since they are low-textured, without the help of temporal consistency term, their data term is dominated by the smooth term and the foreground depth propagates to them. In their turn, they propagates the wrong depth result to their low-textured neighbors.

To solve this problem, the method focuses on preventing the depth propagation from the foreground to the background by adding a background enhancement term into the data term of background pixels around motion. For more specific, as the background of a scene changes slower than the foreground, the intensities of pixels in the foreground do not change much over frames. The detected background of the previous frame, therefore, can be stored and used as the reference to discriminate the background from the foreground. In the method, two types of background maps including background intensity map and background depth map are stored over frames (Figure 20). To reduce the noise created by falsely estimate a foreground pixel as a background one, an exponential filter is applied to background intensity map.

Figure 19. Motion search

$$BI(x,y) = \begin{cases} \alpha BI_{prev}(x,y) + (1-\alpha) I_c(x,y) & if\ d(x,y) < Thres_{bg}\ and\ BI_{prev}(x,y) \neq 255 \\ I_c(x,y) & if\ d(x,y) < Thres_{bg} and\ BI_{prev}(x,y) = 255 \\ BI_{prev}(x,y) & if\ d(x,y) \geq Thres_{bg} \end{cases} \quad (15)$$

$$BD(x,y) = \begin{cases} d(x,y)\ if\ d(x,y) < Thres_{bg} \\ BD(x,y)\ otherwise \end{cases}, \quad (16)$$

Where

$Thres_{bg}$ is the depth threshold to separate the depth of foreground and that of background.

As mentioned above, a background enhancement term is added into the data term to preserve the correct depth of previous frames:

$$E_{data}(d) = \begin{cases} 0 & if\ MS(x,y) = static\ and\ d(x,y) = d_{init}(x,y) \\ 2C\big(x,y,d(x,y)\big) & if\ MS(x,y) = static\ and\ d(x,y) \neq d_{init}(x,y) \\ C\big(x,y,d(x,y)\big) + C_{temporal}\big(x,y,d(x,y)\big) & if\ temporal\ consistency \\ C\big(x,y,d(x,y)\big) + C_{bg_{enhance}}\big(x,y,d(x,y)\big) & if\ background\ enhance \\ C\big(x,y,d(x,y)\big) & otherwise \end{cases} \quad (17)$$

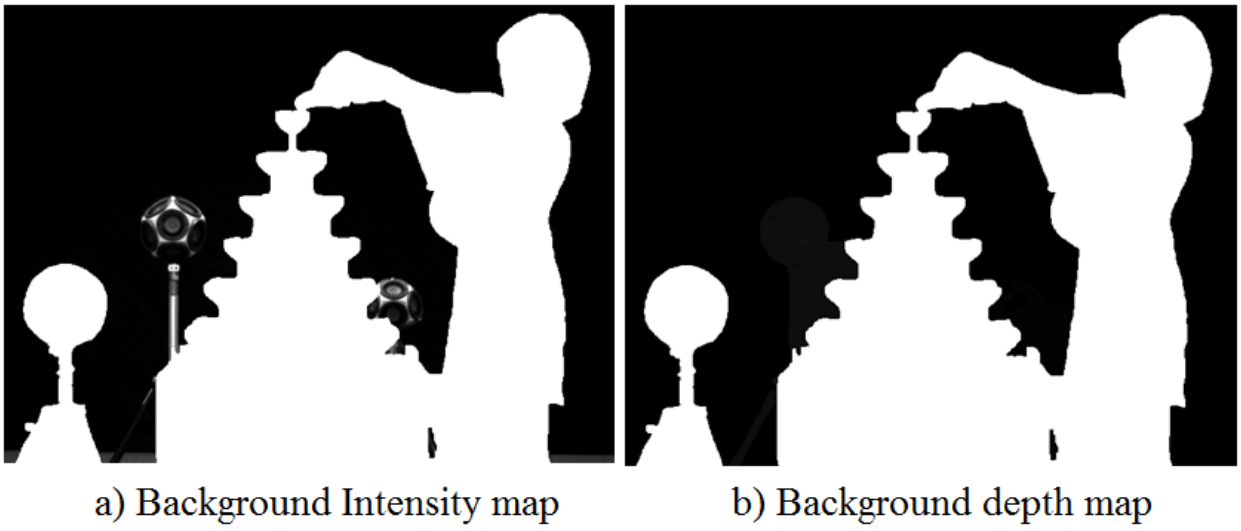
where

temporal consistency:$\sum_{(i,j)\in w(x,y)}\left|I_c(i,j)-I_{cprev}(i,j)\right| < Thres_{motion}$ like (9)

background enhance: not temporal consistency and

$$|I_c(x,y) - BI(x,y)| < Thres$$

If there is the manual static map, it will be used firstly to change the data term. Then, block motion search 16x16 is applied to find the no motion area, which temporal consistency term is used to protect the depth of the previous frame. In detected motion area, intensities of pixels are compared with the stored intensities of pixels of the background intensity map to find the background of sequence and the background depth map is used as the reference for the previous depth.



a) Background Intensity map b) Background depth map

Figure 20. Background Intensity map and Background Depth map

RESULTS AND DISCUSSIONS

4.1. Experiments Setup

As the lack of the resource of the ground truth of Champagne and Pantomime, the experiments to test the result of new method base only the color input sequence. Figure 21 shows the idea of the experiments. The color sequences from camera 38, 39 and 40 are used to estimate the depth sequence of Camera 39; those from camera 40, 41 and 42 are used to estimate the depth sequence of camera 41. Based on the existing depth and color sequences of camera 39 and camera 41, a color sequence from virtual camera 40 is synthesized and compared with that from real camera 40. The Peak Signal Noise Ratio (PSNR) index is calculated at each frame and used as the objective measurement for the quality of depth estimation in these experiments.

$$PSNR = 20\log_{10}\frac{\max_{(x,y)}|I_{origin}(x,y)|}{\sqrt{MSE}}, \tag{18}$$

Where

$$MSE = \sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left(I_{origin}(x,y) - I_{syn}(x,y)\right)^2$$

and I_{origin}, I_{syn} is the original and synthesized images, respectively

m, n is the width and height of both I_{origin} and I_{syn}

“Greater resemblance between the images implies smaller RMSE and, as a result, larger PSNR” [19]. The PSNR index, therefore, measured the quality of the synthesized image. As all experiments used the same synthesize approach, implemented by the reference program of HEVC, the quality of synthesized images shows the quality of depth estimation.

The sequences Champagne, Pantomime and Dog from [8] are used to test in these experiments. In the Champagne and Pantomime tests, the second mode of DERS are used, while the automatic DERS mode is used in the Dog test. DERS with the background enhancement method is compared with DERS without it.

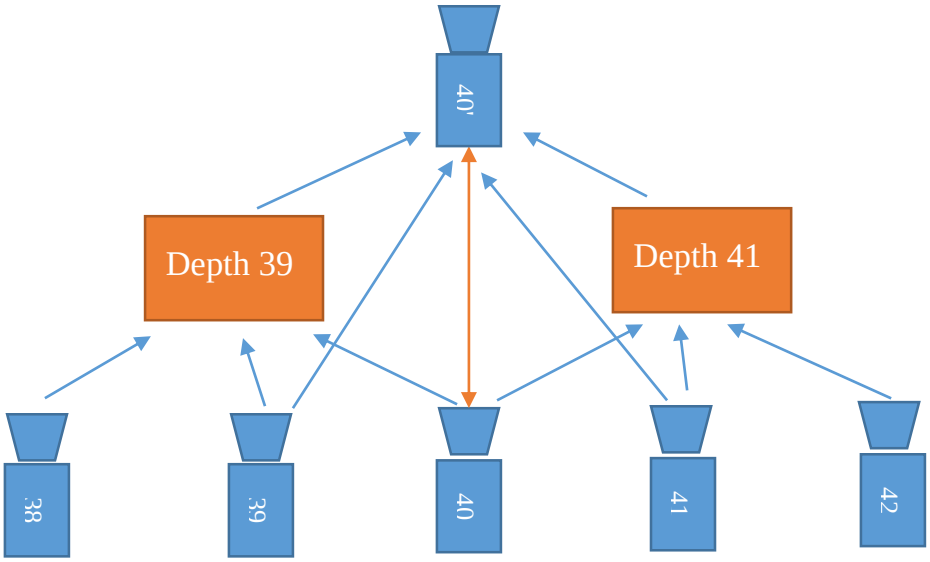
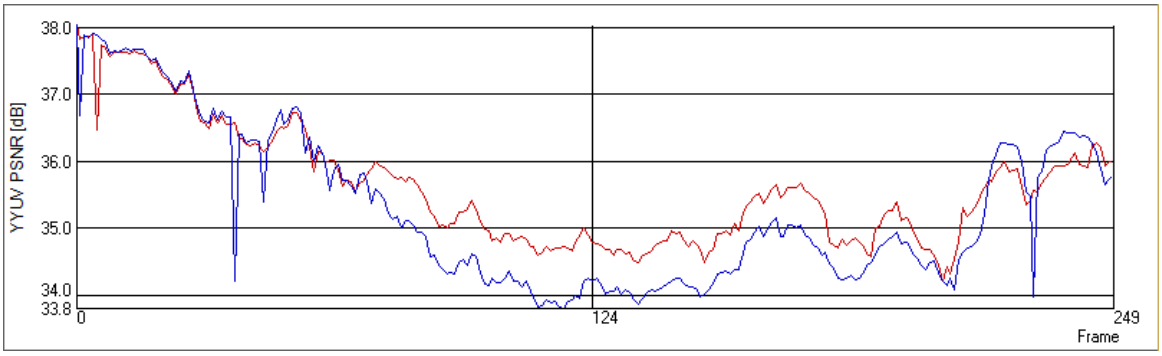


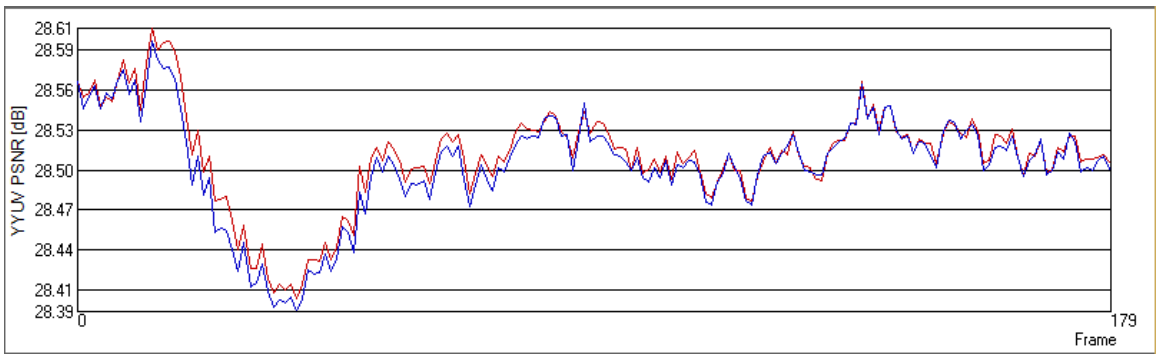
Figure 21. Experiment Setup

4.2. Results

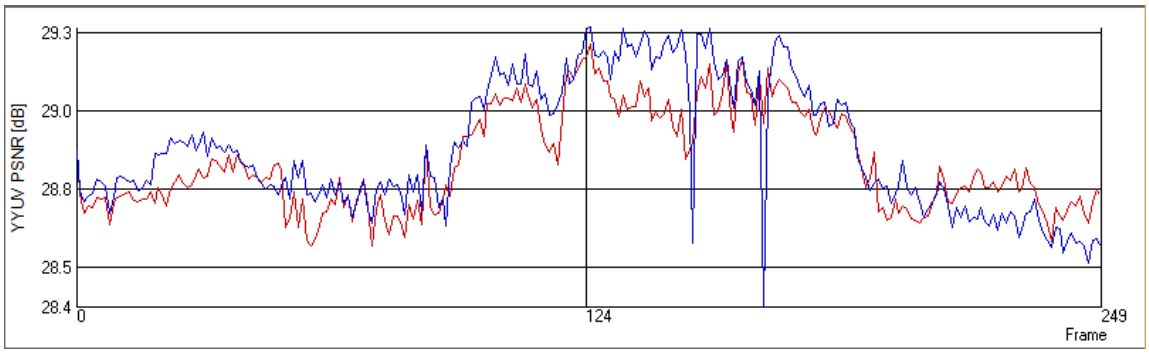
The comparison graphs of Figure 22 and Table 2 shows the results of the tests based on PSNR.



a) Pantomime



b) Dog



c) Champagne

Figure 22. Experimental results. Red line: DERS with background enhancement. Blue line: DERS without background enhancement

Table 2. Average PSNR of experimental results

Sequence	PSNR of original DERS	PSNR of proposed method
Pantomime	35.2815140	35.6007700
Dog	28.5028580	28.5094560
Champagne	28.876678	28.835357

The sequence Pantomime test - the motivation example - shows a positive result with the improvement of about 0.3 dB. In frame to frame comparison between two synthesized sequences from the Pantomime test, it shows that in the first 70 frames, the depth difference between foreground (two clowns) and the low-textured background is not too big (Figure 24.a, b), which makes the two synthesized sequences very resembling. After frame 70th, the difference is large; the propagation of the foreground depth happens strongly (Figure 24.d). The background enhancement method has successfully mitigate this process as in Figure 24.c, which makes the PSNR result increase. However, Figure 24.e shows that the background enhancement cannot stop completely this propagation process but only slow it down. The results from the Dog test show only insignificant improvement in the average PSNR of 0.007 dB. On the other hand, the Champagne test shows a negative result. Although the Champagne sequence has a low-textured background like the Pantomime, it has some features that the Pantomime does not have. Some foreground areas in the Champagne are very similar in color with the background. This leads to the wrong estimation these areas as background areas if we use background enhancement (Figure 23).

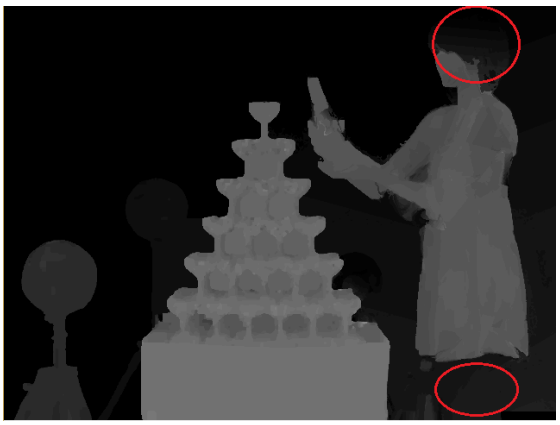
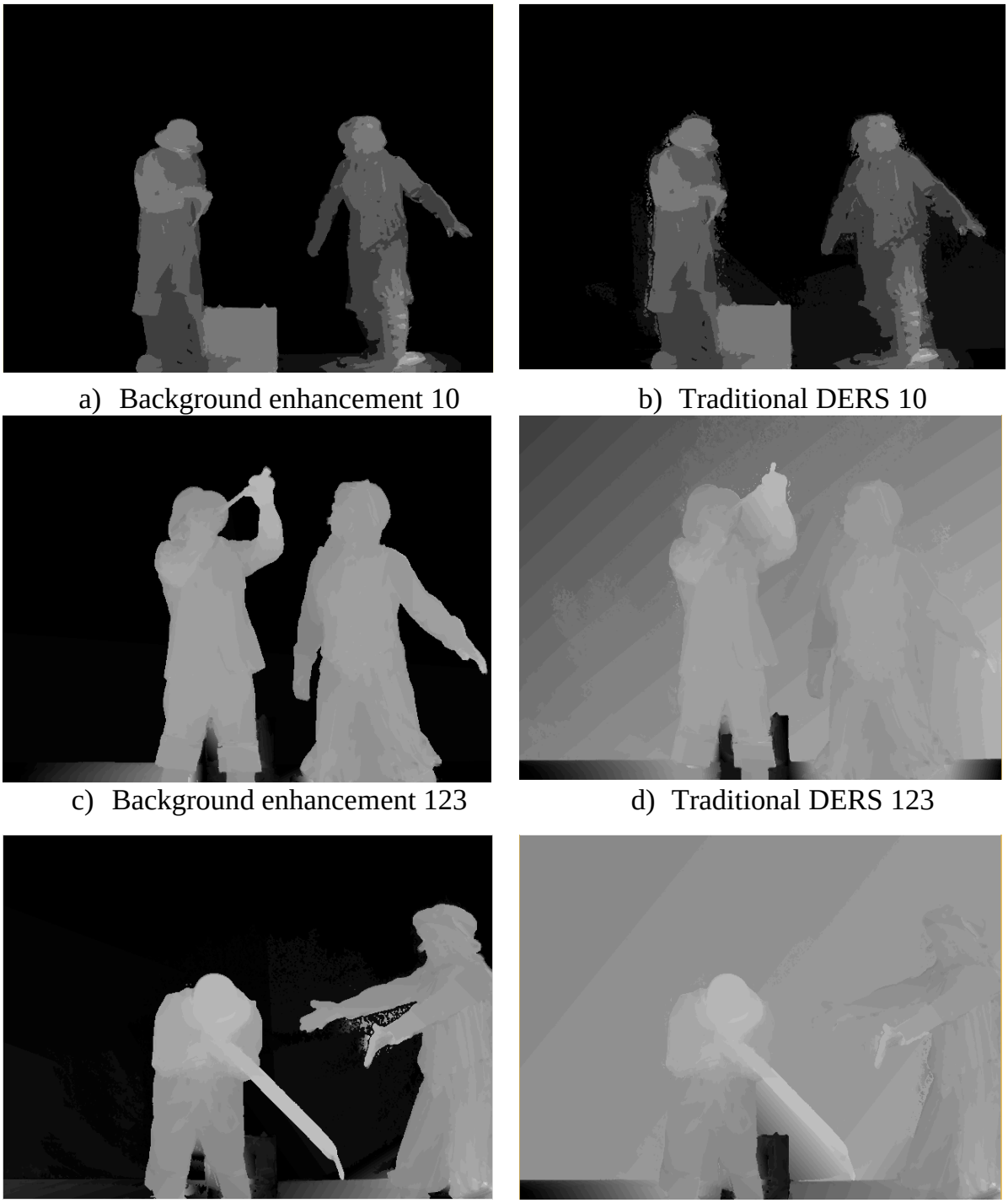


Figure 23. Failed case in sequence Champagne



a) Background enhancement 10

b) Traditional DERS 10

c) Background enhancement 123

d) Traditional DERS 123

e) Background enhancement 219

f) Traditional DERS 219

Figure 24. Comparison frame-to-frame of the Pantomime test. Figure a and b have been processed for better visual effect.

CONCLUSION

In my opinion, Free-viewpoint Television (FTV) is going to be the future of television. However, there is still a long way to get there in both coding and display problems. The solution for multi-view video coding plus depth, in some cases, has helped to solve the problem of coding for FTV. However, it is still required more improvements in this area, especially in the depth estimation as it holds a key role to synthesize views from any viewpoints. MPEG is one of the leading group trying to standardize the Multi-view Video Coding process (including depth estimation) with different versions of reference software like Depth Estimation Reference Software (DERS) and View Synthesis Reference Software (VSRS).

In this thesis, I have given the reader an insightful look into the structure, configuration and methods used in DERS. Moreover, I have proposed a new method called background enhancement to improve the performance of DERS, especially in the case of low-textured background. The experiments have shown positive results from the method in low-textured background area. However, it still has not successfully stopped the propagation of the depth of the foreground to background like the first expectation and has not estimated correctly foreground areas which have color similar to background.

REFERENCES

[1] M. Tanimoto, "Overview of FTV (free-viewpoint television)," in *International Conference on Multimedia and Expo*, New York, 2009.

[2] M. Tanimoto, "FTV and All-Around 3DTV," in Visual Communications and Image Processing, Tainan, 2011.

[3] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima and Y. Mori, "Reference Softwares for Depth Estimation and View Synthesis," in ISO/IEC JTC1/SC29/WG11, M15377, Archamps, April 2008.

[4] M. Tanimoto, T. Fujii and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo," in ISO/IEC JTC1/SC29/WG11 M14888, Shenzhen, October 2007.

[5] M. Tanimoto, T. Fujii and K. Suzuki, "Improvement of Depth Map Estimation and View Synthesis," in ISO/IEC JTC1/SC29/WG11 M15090, Antalya, January 2008.

[6] K. Wegner and O. Stankiewicz, "DERS Software Manual," in ISO/IEC JTC1/SC29/WG11 M34302, Sapporo, July 2014.

[7] A. Olofsson, "Modern Stereo Correspondence Algorithms: Investigation and evaluation," Linköping University, Linköping, 2010.

[8] T. Saito, "Nagoya University Multi-view Sequences Download List," Nagoya University, Fujii Laboratory, [Online]. Available: http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/. [Accessed 1 May 2015].

[9] M. Tanimoto, T. Fujii and K. Suzuki, "Depth Estimation Reference Software (DERS) with Image Segmentation and Block Matching," in ISO/IEC JTC1/SC29/WG11 M16092, Lausanne, February 2009.

[10] O. Stankiewicz, K. Wegner and Poznań University of Technology, "An enhancement of Depth Estimation Reference Software with use of soft-segmentation," in ISO/IEC JTC1/SC29/WG11 M16757, London, July 2009.

[11] O. Stankiewicz, K. Wegner, M. Tanimoto and M. Domański, "Enhanced Depth Estimation Reference Software (DERS) for Free-viewpoint Television," in ISO/IEC JTC1/SC29/WG11 M31518, Geneva, October 2013.

[12] S. Shimizu and H. Kimata, "Experimental Results on Depth Estimation and View Synthesis with sub-pixel precision," in ISO/IEC JTC1/SC29/WG11 M15584, Hannover, July 2008.

[13] O. Stankiewicz and K. Wegner, "Analysis of sub-pixel precision in Depth Estimation Reference Software and View Synthesis Reference Software," in ISO/IEC JTC1/SC29/WG11 M16027, Lausanne, February 2009.

[14] Y. Boykov, O. Veksler and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, November 2001.

[15] M. Tanimoto, T. Fujii, M. T. Panahpour and M. Wildeboer, "Depth Estimation for Moving Camera Test Sequences," in ISO/IEC JTC1/SC29/WG11 M17208, Kyoto, January 2010.

[16] S.-B. Lee, C. Lee and Y.-S. Ho, "Temporal Consistency Enhancement of Background for Depth Estimation," 2008.

[17] G. Bang, J. Lee, N. Hur and J. Kim, "Depth Estimation algorithm in SADERS1.0," in ISO/IEC JTC1/SC29/WG11 M16411, Maui, April 2009.

[18] M. T. Panahpour, P. T. Mehrdad, N. Fukushima, T. Fujii, T. Yendo and M. Tanimoto, "A Semi-Automatic Depth Estimation Method for FTV," The

Journal of The Institute of Image Information and Television Engineers, vol. 64, no. 11, pp. 1678-1684, 2010.

[19] D. Salomon, Data Compression: The Complete Reference, Springer, 2007.

[20] M. Tanimoto, T. Fujii and K. Suzuki, "Reference Software of Depth Estimation and View Synthesis for FTV/3DV," in ISO/IEC JTC1/SC29/WG11 M15836, Busan, October 2008.