

國立交通大學

電子工程學系 電子研究所碩士班

碩士論文

1920x1080@90fps 之深度運算設計與實作

A 1920x1080@90fps Disparity Estimation Design with Edge-oriented

Two Way Dynamic Programming Optimization and Disparity

1896
Propagation

研究生：吳英佑

指導教授：張添烜

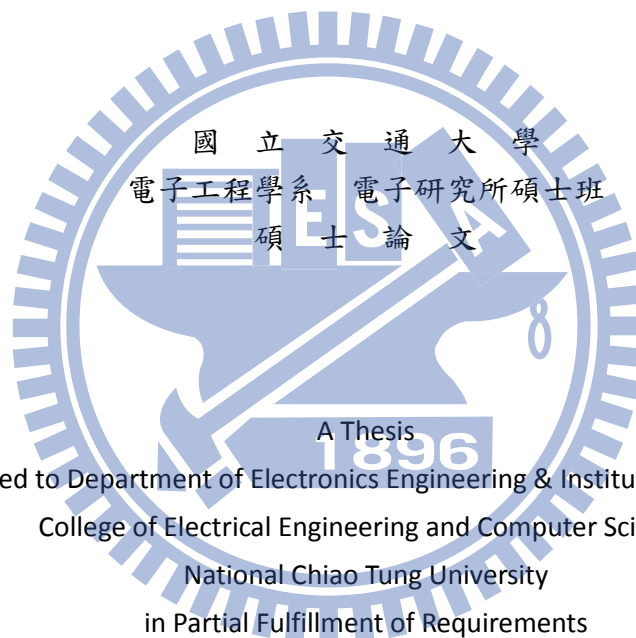
中華民國 一〇一年 十二月

1920x1080@90fps 之深度運算設計與實作

A 1920x1080@90fps Disparity Estimation Design with Edge-oriented
Two Way Dynamic Programming Optimization and Disparity
Propagation

研究生: 吳英佑
指導教授: 張添烜 博士

Student: Ying-Yu Wu
Advisor: Tian-Sheuan Chang



Submitted to Department of Electronics Engineering & Institute of Electronics
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of Requirements
for the Degree of
Master of Science
In
Electrical Engineering
September 2008
Hsinchu, Taiwan, Republic of China

中華民國 一〇一年 十二月

學生: 吳英佑

指導教授: 張添烜

國立交通大學電子工程學系暨電子所碩士班

摘要

隨著立體電視的問世，人們可以藉由立體視訊獲得新的視覺經驗。立體視訊可以立體攝影機擷取，並經由影像處理技術運算後，可支援多視角與自由視點之立體電視應用。在立體視訊的處理中，視差估測為最重要的技術之一。視差估測可產生拍攝場景之視差圖，可用於虛擬視角視訊的合成。動態影像壓縮標準組織的立體視訊編碼團隊已提出目前最先進視差估測演算法。其演算法可針對立體電視的應用產生高品質的視差圖，但因採用圖形切割演算法導致高運算複雜度與低平行運算的問題。特別對於高畫質視訊，其問題更為嚴重。

為解決以上問題，本論文提改良過的雙向動態規劃演算法，利用參考邊界資訊的遮蔽處理、方向性投票機制、以及利用邊界資訊處理不同時間嚇得像素穩定性，以達到高品質書度運算輸出之需求。另外本文亦提出深度傳輸之演算法，可以有效降低運算時間至50%以上。另一方面，針對超大型積體電路設計，本文提出之硬體架構可以合成出以下數據之電路：1920x1080@90fps，另外電路的閘數量為2,325K，使用UMC 90nm CMOS製成合成。

A 1920x1080@90fps Disparity Estimation Design with Edge-oriented Two Way Dynamic Programming Optimization and Disparity Propagation

Student: Ying-Yu Wu

Advisor: Tian-Sheuan Chang

Department of Electronics Engineering & Institute of Electronics

National Chiao Tung University

Abstract

Disparity estimation is one of the most interesting and important research topics in the field of stereo TV application. Accurate estimation of disparity can significantly improve the visual experience on the stereo image but at the expense of noticeable computational complexity consumptions.

In this thesis, several techniques are proposed to improve the accuracy of estimated disparity results at a low memory cost. The edge detection algorithm is first adopted in the proposed algorithm to derive the important image content features and edge information for making the upcoming disparity estimation process gets more precise results. Afterwards, the proposed disparity propagation will take the edge information both from vertical and horizontal direction into account for deciding whether the disparity should be propagated from the edge area to the texture-less area. After the disparity propagation phase, the un-propagated pixels will be treated by our proposed dual-way dynamic programming method for determining their disparities. In our proposed dual-way dynamic programming algorithm, the edge information will be taken into account as the energy minimization factor which will affect the results of the estimated disparity. In addition, several post processing techniques including occlusion handling, directional region voting, and edge-based temporal consistency are also adopted in this thesis to further improve the estimated disparity results with considering edge information.

Simulation results demonstrate that our proposed disparity estimation algorithm not only improves the accuracy of the estimated disparity but also achieves less computational complexity consumptions and memory buffer requirements. On average, our proposed algorithm can achieve 34.48dB PSNR and reduce average 53.08 % of computation cost compared to the conventional dynamic programming method. Finally, the proposed algorithm is implemented in hardware form at 1920x1080@90fps and the synthesized gate count of our design is only 2,325K by using 90nm CMOS technology.

誌謝

從大學到研究所，交大給予我許多成長與回憶。在此，藉由碩士論文的完成以感謝所有的人。首要感謝的是指導老師張添烜教授，不論在研究方法、論文撰寫、為人處世皆給予我耐心的指導與建議。接著要感謝趙家佐教授，在我大學三年級擔任指導教授指導專題研究。另外也感謝其他口試委員，包含王聖智教授、蔡淳仁教授，願意撥空給予指導。

工程四館 427 實驗室是我碩士班在交大停留最久的地方，首要感謝的是曾宇晟學長，教導我良好的研究方法與態度，並引領我進入碩士論文的研究題目。也感謝實驗室學長國龍和易群，在研究生涯中與我分享研究及生活。接著要感謝實驗室的同學們：亮齊、孟勳、克嘉、輔仁、兆傑、珊榕、碩文。大家在生活及研究上互相討論學習，並營造實驗室和樂的氣氛。因為有你們實驗室總是歡笑不斷。

最後要感謝我的家人及朋友，感謝我的父母一路以來的支持及栽培，終於順利完成了求學之路，另外也要感謝我的好朋友世昌在這一路上的支持。此外也感謝雨濃在這一路上的陪伴及支持。



Table of Contents

摘要.....	III
I. INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.2. MOTIVATION.....	2
1.3. CONTRIBUTION	2
1.4. ORGANIZATION	49
II. RELATED WORKS.....	3
2.1. DISPARITY ESTIMATION.....	3
2.2. GENERAL ALGORITHM FLOW.....	6
III. PROPOSED DUAL-WAY DYNAMIC PROGRAMMING ALGORITHM FOR DISPARITY ESTIMATION.....	25
3.1. OVERALL FLOWCHART.....	25
3.2. COST CALCULATION PHASE.....	26
3.3. DISPARITY ESTIMATION PHASE.....	30
3.4. DISPARITY REFINEMENT PHASE.....	40
3.5. SIMULATION RESULTS.....	48
IV. HARDWARE IMPLEMENTATION AND RESULTS.....	53
4.1 OVERALL ARCHITECTURE.....	53
4.2 FIRST STAGE DESIGN.....	54
4.3 SECOND STAGE DESIGN.....	57
4.4 THIRD STAGE DESIGN.....	58
4.5 IMPLEMENTATION RESULTS.....	61
V. CONCLUSION AND FUTURE WORK.....	62
5.1 CONCLUSION.....	62
5.2 FUTURE WORK.....	62

I. Introduction

1.1. Background

In the research field of computer vision, the stereo matching [1]-[53] is one of the most active and interesting issues. The stereo matching techniques try to analyze the stereo images pair by pair. Afterward, the displacement of corresponding pixel pair existing in both images are estimated for deriving the depth information of objects in the scene. Here, the displacement is measured in pixel unit and we usually called the Disparity. The disparity values are usually between certain rang we usually called Disparity Range and the disparities of all image pixels are grouped to form the Disparity Map. Finally, the disparity map is the target output of stereo matching and it will be used in the stereo related visual processing. Fig. 1. 1 shows an example of *Teddy* test benchmark image. In this figure, Fig. 1. 1(a) and Fig. 1. 1(b) are the images of left and right view, respectively. In addition, the Fig. 1. 1(c) is the ground truth disparity map of left image which is visualized as grayscale intensities. In the disparity map, the brighter grayscale means that the object is much close to the stereo cameras. In other words, the disparity map includes the depth information of each pixel in somewhere. As a result, if we obtain the depth information by using the stereo matching, we will be able to derive the 3D information and reconstruct the 3D scene by means of triangulation.



(a) (b) (c)

Fig. 1. 1. An example of disparity map (a) Left view, (b) Right view, and (c) Ground truth disparity map of left image

1.2. Motivation

Many disparity estimation algorithms have been developed in computer vision for different applications, such as 3DTV, gesture recognition, robot, 3D interactive interface, and etc. Both dynamic programming and belief propagation are approaches that produce better result than others. Computational complexity and memory usage of scan-line based dynamic programming is much lower than the belief propagation. But traditional dynamic programming still requires big buffer size due to buffering information of the path table for tracing back the results of estimated disparity. Therefore, the buffer size increases rapidly with the incensement on frame width.

Motivated by the problems in the dynamic programming disparity estimation algorithm, the goal of this thesis is to develop a new scan-line based dynamic programming disparity estimation kernel that could not only generate high quality disparity maps, but also achieve the throughput of 60 frames/s for the HD1080p resolution to satisfy the requirement of high definition 3DTV applications.

1.3. Contribution

To achieve the above goals, this thesis proposes several techniques to reduce the computational complexity and improve the quality as well. The main contributions of the proposed algorithm are in several parts as mention below. First, the proposed disparity propagation algorithm saves more than 50% computation complexity generally in most of the test contents. Second, to solve the memory cost problem of traditional dynamic programming, the thesis

proposes an alternative which is named “Dual Way Dynamic Programming”. This algorithm solves the memory usage problem effectively and also improves the quality of the output disparities by computing energy function in dual directions. Third, we propose several creative post processing techniques such as “Edge-Based Occlusion Handling”, “Directional Regional Voting”, and “Edge-based Temporal Consistency” to solve problems such as occlusion on boundaries, bad influence caused by pixels in different objects in the region, and the flickering problem during video playback.

Our proposed algorithm can be implemented up to 1080P@90fps with UMC 90nm CMOS technology which generates two disparity maps and uses 2335K gate counts and will fit the HD 3DTV trend.

II. Related Works

2.1. Disparity Estimation

Disparity estimation is one of the key techniques which extracts the disparity information from source images and produces the disparity map for each image in 3DTV applications. Afterward, the extracted disparity map could be used to present the relative distance of objects in scene. In addition, the disparity map could be further adopted to obtain virtual-view images. The approach for disparity estimation depends on the number of input image views. The traditional single-view image uses the 2-D to 3-D conversion technique while the two-view and multiple-view images use the stereo correspondence techniques. The traditional 2-D to 3-D conversion technique identifies the disparity map from different disparity cues, such as texture, defocus, vanishing point, and etc. [49], [50], and [51]. On the other hand, the stereo correspondence techniques find the pairs of correspondences for deriving disparity maps.

The inherent constraint of epipolar geometry could lead to the correspondence

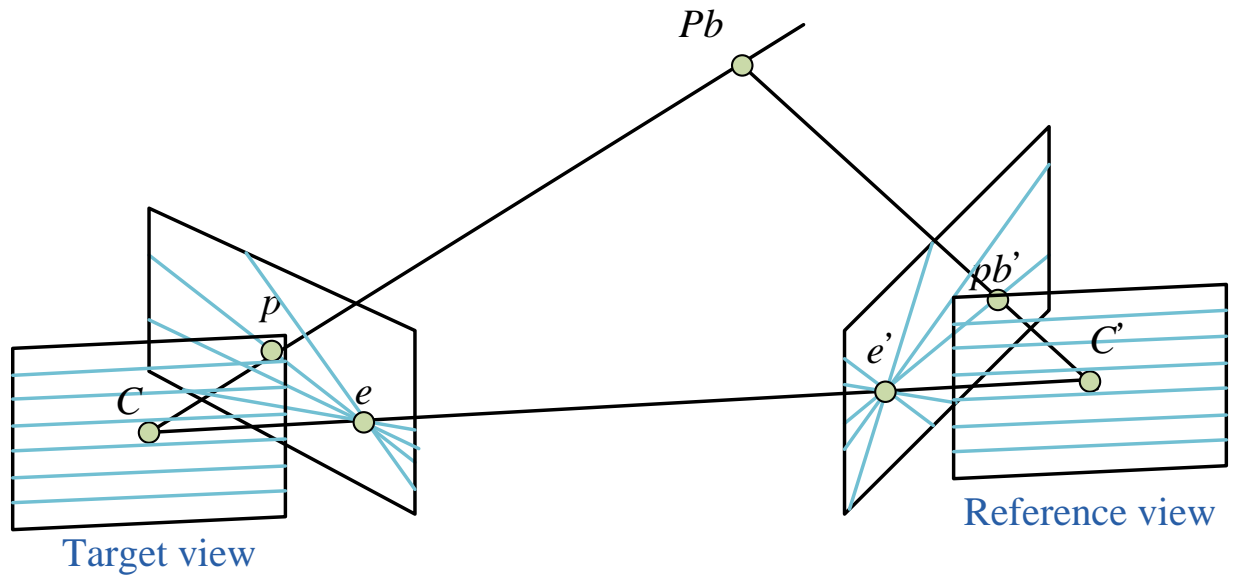


Fig. 2. 2 Rectification for image planes

For the rectified image planes, the relationship between depth and disparity of a correspondence pair is shown in Fig. 2. 3. In this figure, we can find that the object point Pb is captured by the two cameras at the viewpoints of C and C' and projected onto the correspondence pair on the epipolar line. The correspondences are located at the coordinates of X and X' based on their camera centers. Therefore, if we are able to estimation the disparity $X-X'$ when given the focal length f and the baseline B of the cameras, the object depth Z can be acquired by

$$Z = \frac{f \times B}{X - X'}$$

As a result, the disparity estimation tries to find out the correspondence pair and uses their x-coordinates to derive the disparity of depth value for each pixel.

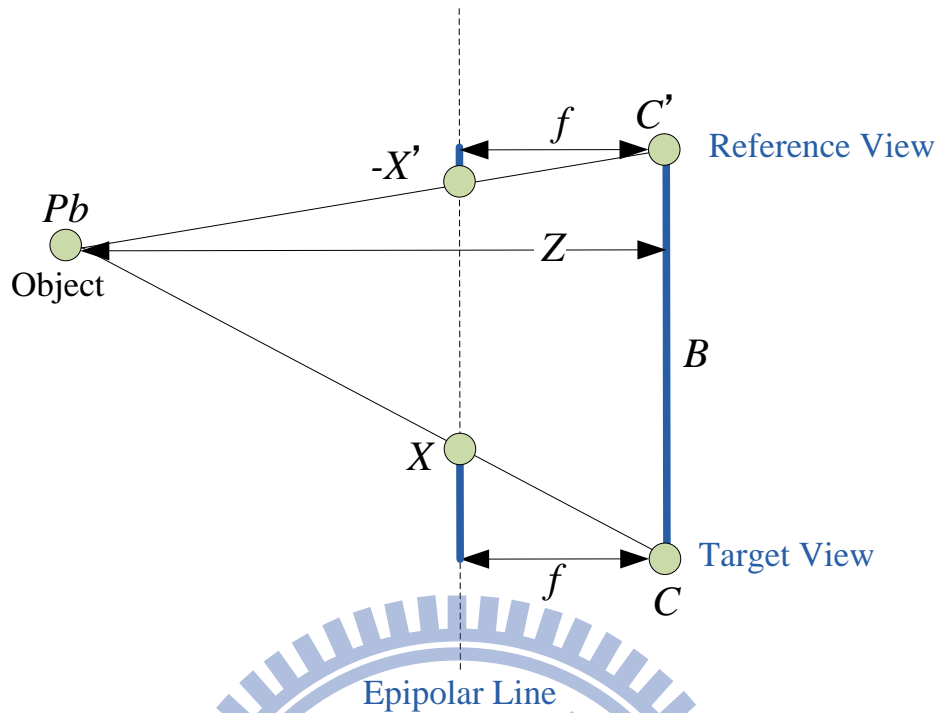


Fig. 2. 3 Relationship between disparity and depth for a correspondence pair

2.2. General Algorithm Flow

Fig. 2. 4 shows a general framework for disparity estimation algorithms proposed by *Scharstien* and *Szeliski* [52]. Two images are first obtained and rectified to be the inputs and the expected result is the disparity map in this frame work. However, the disparity estimation can be roughly classified into two categories: local approach and global approach [52] and [53] in this framework. In the category of local approach, it only consists of the matching cost calculation and the cost aggregation. However, the optimization operation is additional executed for global approach. Finally, the disparity map is refined by the last disparity refinement step which is an optional process for computing fractional disparity and other post-processing. The literatures of each step inside the general framework are briefly reviewed as follows.

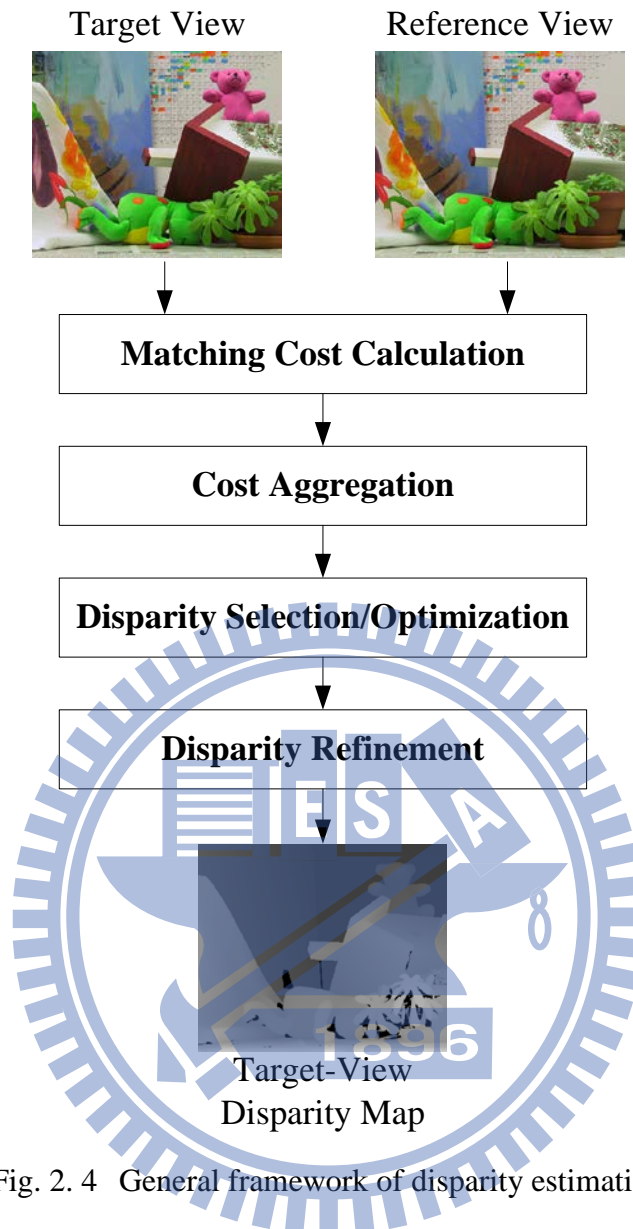


Fig. 2. 4 General framework of disparity estimation

Fig. 4.1. Matching Cost Calculation

To find the best correspondence pair, the matching cost is an essential quantitative evaluation. Fig. 2. 5 exhibits an example to illustrate the calculation of matching cost. In this figure, multiple reference pixels are marked as the correspondence candidates and all their matching costs have been computed accompanied a target pixel. However, the relation of nearest and farthest objects in

scene is recognized as disparity range DR and it will be used to represent the number of correspondence candidates. As a result, DR matching costs would be produced by the target pixel. In order to find out the overall disparity map, all matching costs of all target pixels have to be calculated and all calculated matching costs form a disparity image space. Fig. 2. 6 shows a disparity image space which contains the spatial dimensions X , Y and disparity dimension d . Overall, this disparity image space consumes $H \times W \times DR$, where H and W are the frame height and width, memory space to store the all matching costs of entire frame.



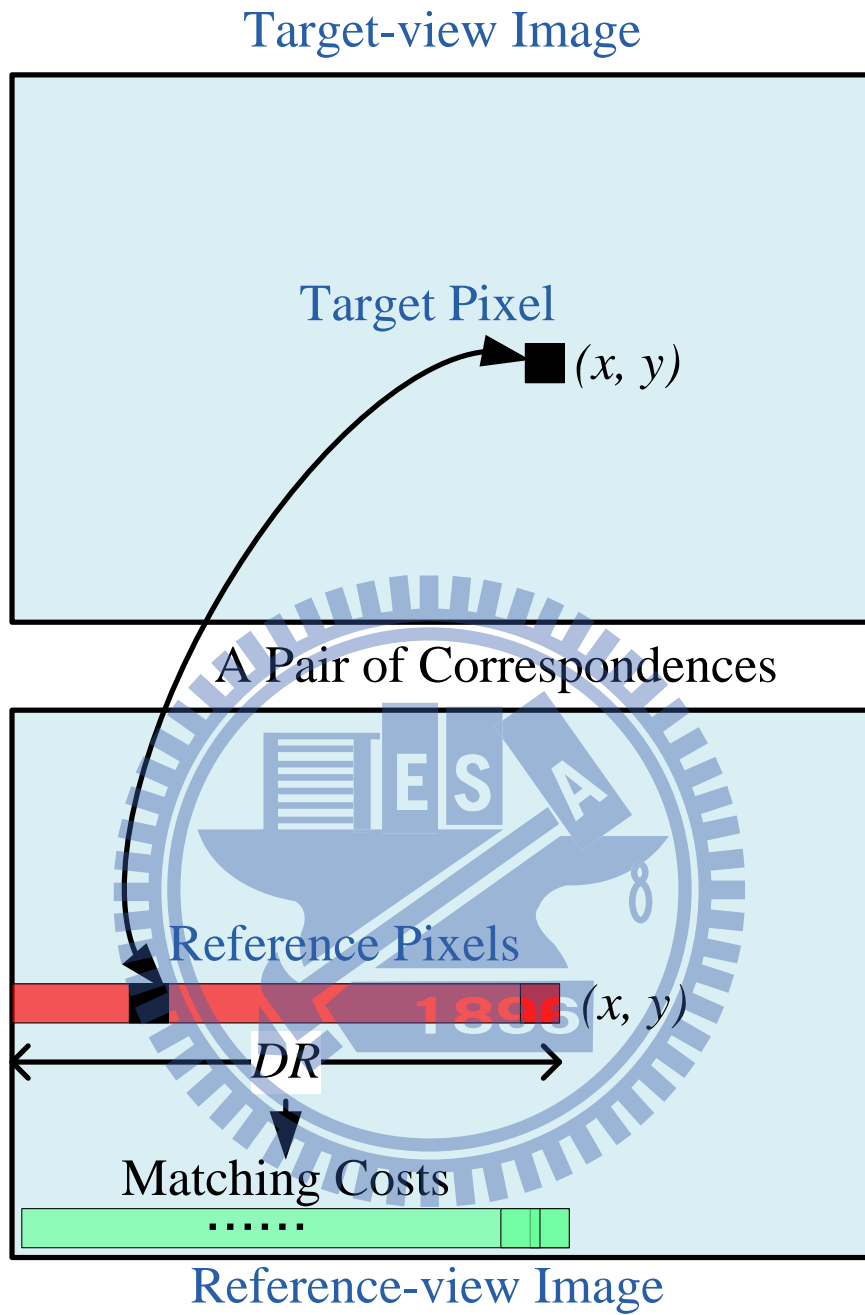


Fig. 2. 5 Matching costs of a target pixel and its correspondence candidates

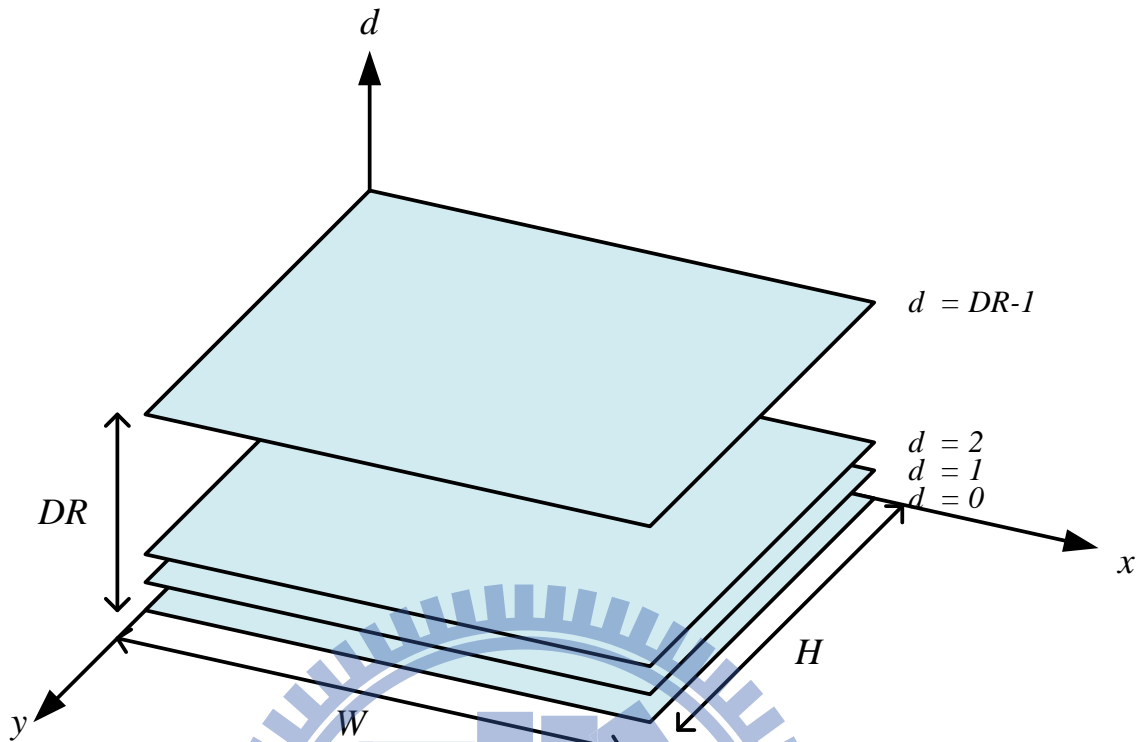


Fig. 2. 6 Matching costs of a target pixel and its correspondence candidates

There are many match measurements [3]-[52] as listed in Table 2-1 could be used to compute the cost disparity image space. These match measurements could be classified into pixel based and block based approach. For the pixel based approach, the absolute difference (AD) and the square difference (SD) are used for computing the matching costs by considering a target and reference pixel. To eliminate the sampling sensitivity [1], the half pixels could be further considered for pixel dissimilarity measurement. On the other hand, instead of using a target and reference pixel to compute the matching cost, a target and reference pixel block is used to compute the block based matching cost as Fig. 2. 7 shown. In addition, the statistical approach called normalized cross correlation reduces the sensitivity of radiometric gain and bias by using the block mean and variance. The Rank derives the rank value

of pixel color by transformation and the rank values are adopted for computing the matching costs. On the other hand, the Census transforms the pixel intensity into census bitstream consisting of the intensity comparison results between the center pixel and the support pixels. Afterward, the Hamming distance is calculated to derive the matching cost of two census bitstreams. In summary, since the Rank and Census try to transform the original pixel from color to another domain, their ability to resist the radiometric distortion between views would be much better.

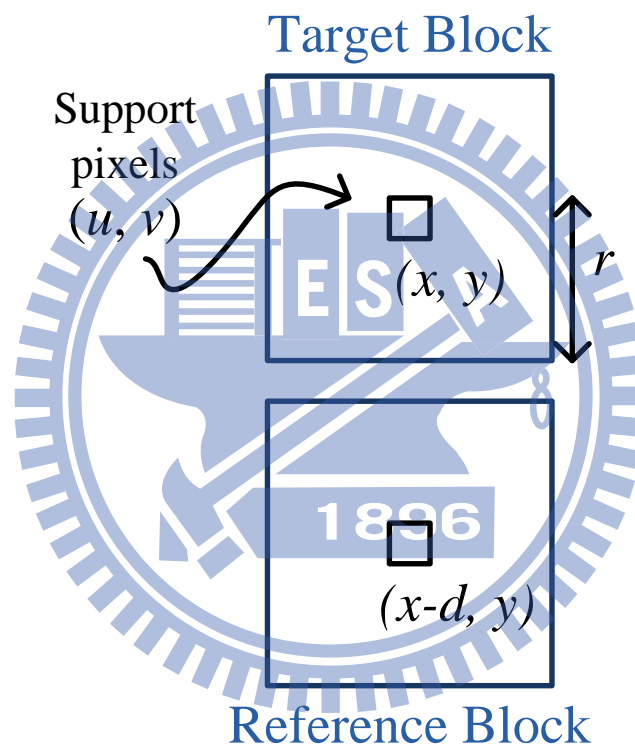


Fig. 2. 7 Block based matching cost calculation

Table 2-1 Different matching cost measurements

Block-based	
Normalized Cross Correlation (NCC)	$\frac{\sum_{\substack{ x-u \leq r \\ y-v \leq r}} [I_{tar}(u, v) - \bar{I}_{tar}][I_{ref}(u - d, v) - \bar{I}_{tref}]}{\sqrt{\sum_{\substack{ x-u \leq r \\ y-v \leq r}} [I_{tar}(u, v) - \bar{I}_{tar}]^2 [I_{ref}(u - d, v) - \bar{I}_{tref}]^2}}$
Rank	$ I'_{tar}(x, y) - I'_{ref}(x - d, y) ,$ <p>where $I'(m, n) = \sum_{ m-u \leq r, n-v \leq r} I(m, n) > I(u, v)$</p>
Census	$Hamming(I'_{tar}(x, y), I'_{ref}(x - d, y)),$ <p>where $I'(m, n) = bitstream_{ m-u \leq r, n-v \leq r}(I(m, n) > I(u, v))$</p>
Pixel-based	
Absolute Difference (AD)	$ I_{tar}(x, y) - I_{ref}(x - d, y) $
Square Difference (SD)	$(I_{tar}(x, y) - I_{ref}(x - d, y))^2$
Pixel Dissimilarity Measure (PDM)	$\min\{ I_{tar}(x, y) - I_{ref}(x - d, y) , I_{tar}(x, y) - I_{ref}^+ , I_{tar}(x, y) - I_{ref}^- \}$ <p>where I_{ref}^+ and I_{ref}^- are the neighboring half pixel of $I_{ref}(x - d, y)$</p>

Fig. 4.2. Cost Aggregation

The main purpose of cost aggregation is tried to gather the neighboring pixel costs in a window for center pixel for further processing usage. The assumption behinds of cost aggregation is that the neighboring pixels tend to have the same disparity and gathering the matching costs from neighbors could be able increase the reliability of matching cost. Therefore, the neighboring costs are accumulated in the cost aggregation step for the center pixel by the following equation,

$$C_{aggr}(x, y, d) = \frac{\sum_{(u,v) \in win(x,y)} C(u, v, d) \times Waggr(u, v)}{\sum_{(u,v) \in win(x,y)} Waggr(u, v)}$$

where C is the initial matching cost and C_{aggr} is the aggregated matching cost. In this equation, each initial cost $C(v, u, d)$ in an aggregation window with window size r is accumulated with the weight $W_{aggr}(u, v)$ for the target cost $C_{aggr}(x, y, d)$. In addition, the accumulated value is normalized by the sum of weights. The computational complexity of this step is $O(H \times W \times DR \times r^2)$ proportional to the aggregation window size.

Fig. 2. 8 shows different cost aggregation methods with different weighting distribution. The uniform weight as shown in Fig. 2. 8(a) contains constant weight and fixed r for every support pixel. However, this uniform weight suffers from the problem of over-blurred disparity map for small objects with too large r and disparity map incorrectness for textureless regions with too small r . Therefore, to receiving better disparity result, dynamically adjusting r according to image content as shown in Fig. 2. 8(b) is a good way to do that. The Gaussian weight approach Fig. 2. 8(c) which tries to make the pixels near window center has higher weighting is another commonly used way for deciding the weighting for cost aggregation. However, the disparity accuracy could not be achieved better due to the fixed window shape such square or circle.

To adaptively change the window shape, the 8-direction or 4-direction configuration as shown in Figure Fig. 2. 8(d) is used in the adaptive polygon weight approach [4] and [5] to fit the object shape. And then, the multiple cross lines concept as shown in Figure Fig. 2. 8(e) is adopted in the cross-based weight approach [6] to fit the object shape. In these two methods, the support region is grown from the window center until the dissimilar pixel has been encountered by the support region boundary. Unfortunately, there two methods can be not performed well for the images with

highly texture content due to their continuous support regions.

However, the above mentioned problems could be able to be solved by the adaptive support-weight (ADSW) approach [7] since all support pixels are considered and their weights are decided by the bilateral filter kernels. The weights of ADSW are defined as

$$W_{aggr}(u, v) = W_{tar}(u, v) \times W_{ref}(u - d, v)$$

where W_{tar} is the weight from target-view window and W_{ref} is the weight from reference-view window. The weights of W_{tar} and W_{ref} can be computed by the bilateral filter kernels listed below,

$$W(u, v) = f(\|(x, y) - (u, v)\|)g(\|I(x, y) - I(u, v)\|)$$

where f is the spatial kernel with the position distance and g is the range kernel with the color distance. As a result, the aggregation weight could be large either the support pixel is near the center pixel or the support pixel is similar to center pixel with the help of two kernels.

Compared to the adaptive polygon weight and cross-based weight approaches, the aggregation weight of the adaptive support-weight shown in Fig. 2. 8(f) could fit object shape better for highly texture regions but at the expensive of significant high computational complexity requirement. However, the high computational complexity issues can be addressed by the integral histogram approach [8], the iterative aggregation with small window approach [9], and the data reuse approach in VLSI design [10]. In overall, by using the well-defined weights, the aggregation cost step can produce more reliable matching cost C_{aggr} which will be very helpful for the upcoming disparity selection and optimization.

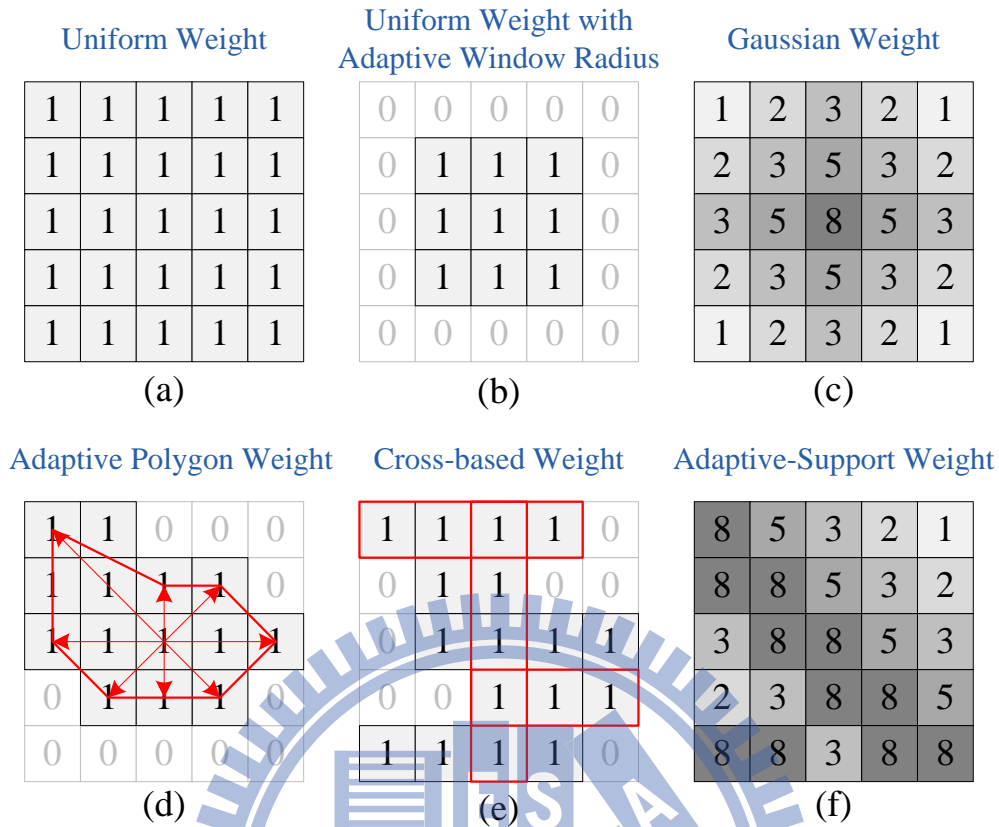


Fig. 2. 8 Different cost aggregation methods (a) uniform weight, (b) uniform weight with adaptive window size, (c) Gaussian weight, (d) adaptive polygon weight, (e) cross-based weight, and (f) adaptive support-weight

Fig. 2. 9

Fig. 4.3. Disparity Selection/Optimization

After the initial costs have been aggregated, the disparity map could be computed by two optional methods. The most common and simple one is the winner-take-all manner (WTA) which decides the disparity result directly by determining the minimum cost reference pixel as the best correspondence for each target pixel. Another disparity optimization approach takes the aggregated costs of entire frame for computing the disparity map through the energy minimization. Literature [48] demonstrated that the latter one can derive more precise disparity maps via the evaluation results.

Some techniques such as dynamic programming (DP), graph-cut (GC), and belief propagation (BP) are the commonly adopted for disparity optimization. In one word, the main concept behinds these disparity optimization techniques are to transform the disparity optimization problem into the energy minimization problem. The energy function could be generally formulated by

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

where E_{data} refers to data term for penalizing the dissimilarity of a correspondence pair and E_{smooth} is smoothness term to penalizing the disparity inconsistency of two neighboring pixels. In addition, d stands for a selected disparity set for entire frame. In one word, a disparity set d is attempted to be found through the approach of minimizing the total energy E that the optimization technique provided.

The principle of some well-known optimization techniques are briefly described below.

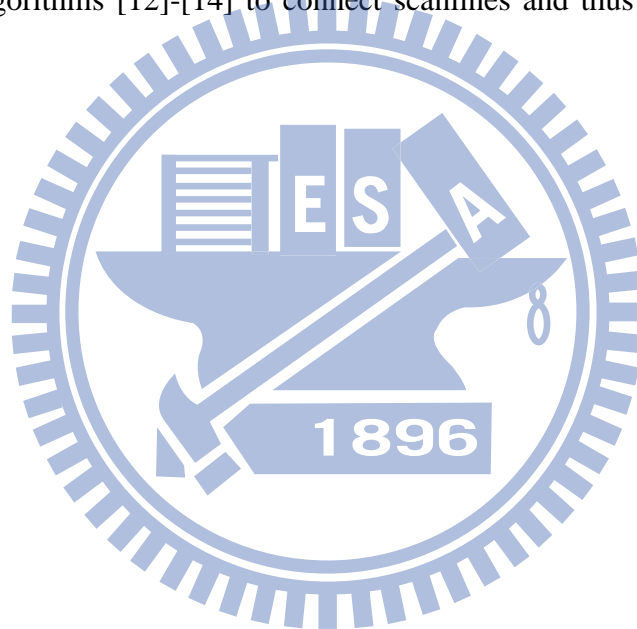
(1) Dynamic Programming

The DP algorithm is a very well-known optimization algorithm which can be used in disparity estimation by mapping the disparity estimation into finding the shortest path problem. In DP, the optimization process is executed in a row by row manner for finding the optimal results.

Fig. 2. 9(a) shows the illustration to demonstrate how the shortest path problem can be solved by DP optimization technique. In this figure, the position of node is corresponding to the coordinate in the $x-d$ plane and the shortest path will be from x of 0 to $W-1$. The path should be suffered from the matching penalty and smoothness penalty on a node and an edge, respectively. During the DP optimization process, two steps called forward accumulating and backward tracing are executed in order to find out the path with minimum penalty. In the first step, the penalties are accumulated in forward direction to find out the moving path for each node as Fig. 2. 9(b) shown.

Afterward, the backward direction tracing as shown in Fig. 2. 9(c) is executed to find the minimum penalty path with the help of the moving direction map that the forward accumulating step produced.

However, the most critical issue caused by the DP technique is that the streak artifact in the disparity map due to the row by row processing mechanism. To eliminate the streak artifact problem, literature of *Ohta* and *Kanade* [11] performed the DP in a 3-D space which consists both of the original intra-scanline and the additional inter-scanline space. In addition, the tree structure has been used in the tree-based DP algorithms [12]-[14] to connect scanlines and thus remove the streak artifacts.



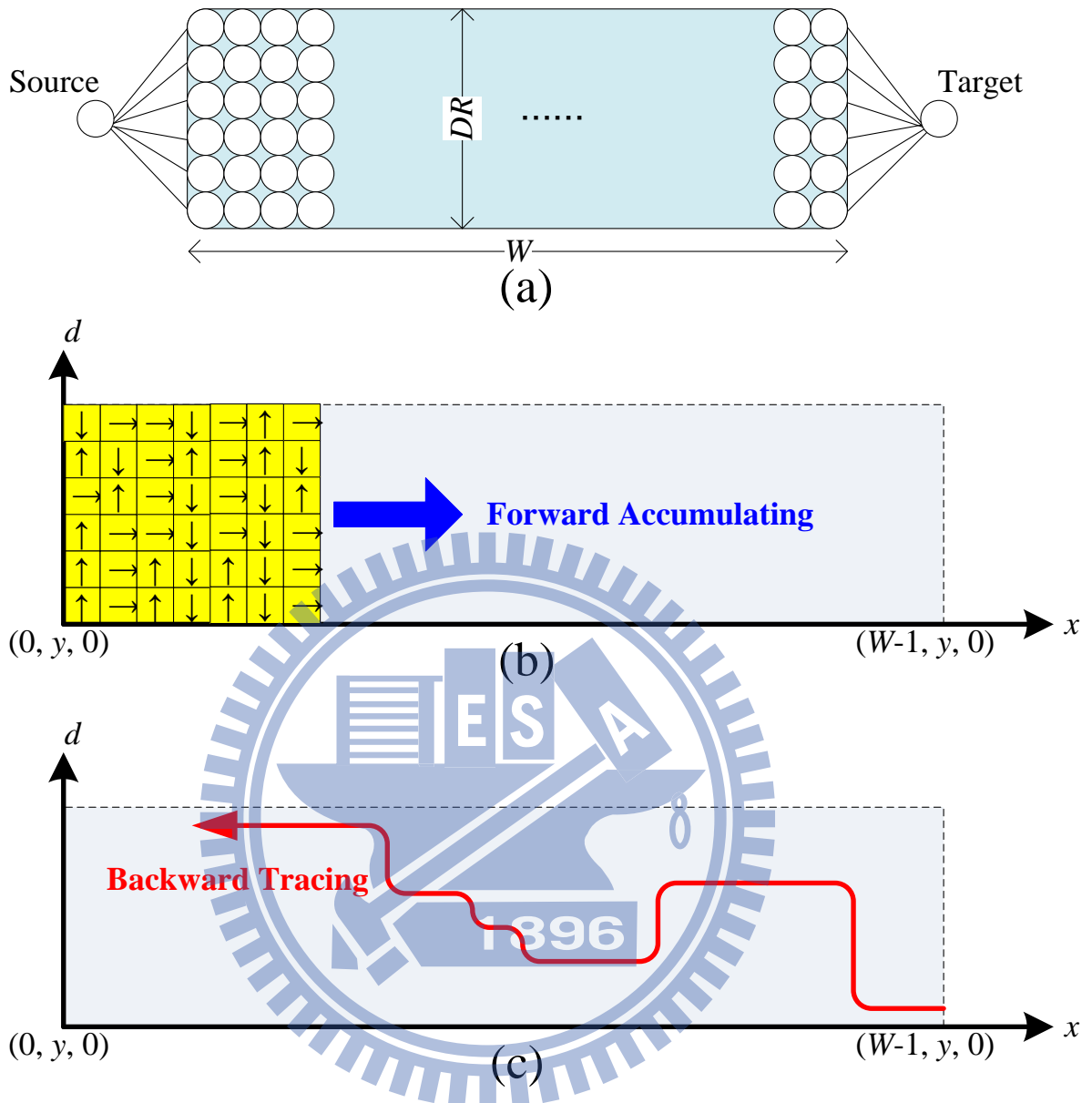


Fig. 2. 10 Illustration of dynamic programming optimization technique (a) graph model in DP approach, (b) forward accumulating, and (c) backward tracing

(2) Graph-Cut

Converting the disparity selection problem into the max-flow/min-cut problem [15] is the key concept of GC optimization technique. In addition, the associated optimization algorithms can be adopted as well for generating more accurate disparity

maps. Fig. 2. 10 shows an example to illustrate the min-cut/max-flow for disparity estimation where there are $H \times W \times DR$ nodes with 6-connected node grid. The well-defined matching cost and smoothness cost on each edge can be regarded as pipes with different flow quantities due to different costs. In this illustration, the water would be flowed from the source to sink node through the pipes. The terms of min-cut and max-flow respectively stand for a cut surface cross edges that has the minimum flow and the allowed maximum flow from the source to the sink. In other words, the problems of min-cut and max-flow are equivalent in somehow. As a result, the disparity map can be obtained directly via the resultant cut surface.

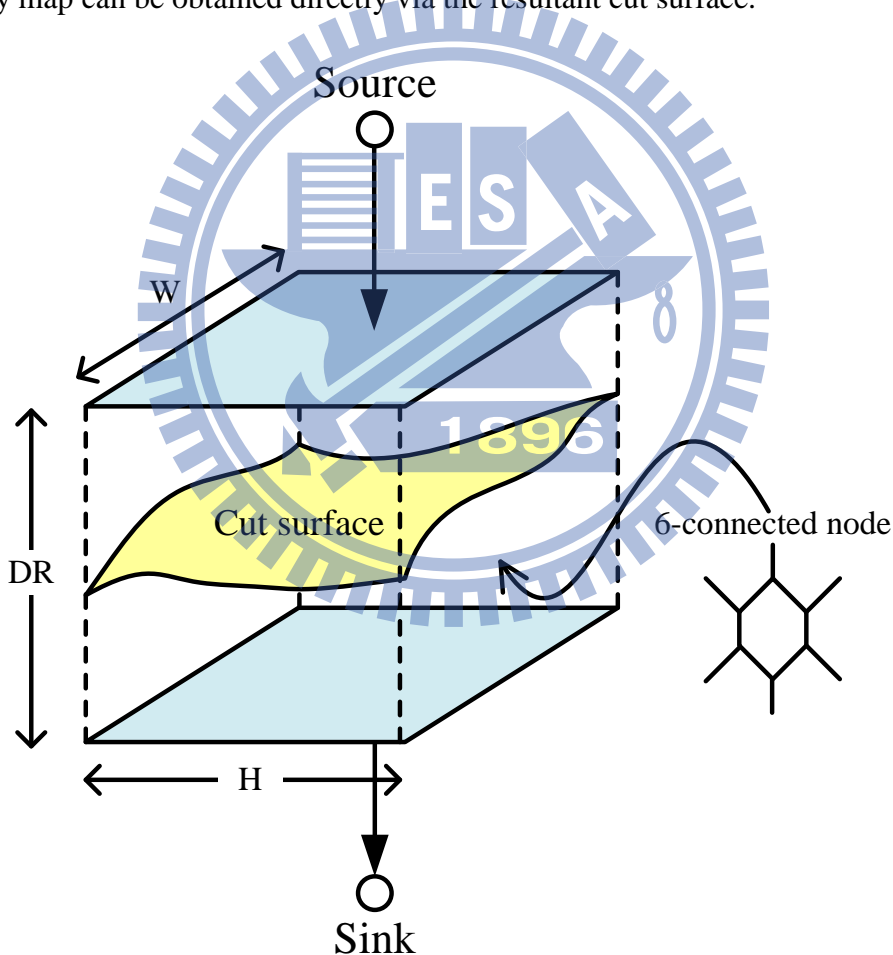


Fig. 2. 11 Illustration of graph-cut optimization technique

The widely used optimization techniques for solving the min-cut/max-flow

problem are the push-relabeling [16] and the augmenting path [17] and their computational complexities are highly depended on the number of label candidates (i.e. disparity range DR in disparity estimation). However, the large disparity range leads to these optimization techniques suffer from extremely high computational complexity problem.

Literatures of swap method [18] and an efficient augmenting path [19] have been proposed by *Boykov* to reduce the computational complexity of GC. The optimization process was performed isparity by disparity in swap method and each iteration only considers one new disparity. In addition, the literature of *Chou et. al.* [20] proposed a fast algorithm to predict the disparities by early skipping the partial optimization process based on the swap method. On the other hand, the computational speed of the push-relabeling approach depends on the processing order on nodes. As a result, *Checkassky and Goldberg* [21] proposed a highest-label order which can achieve more efficient computation than that of the typical FIFO order. In addition, the block-based graph cut algorithm was proposed by *DeLong and Boykov* [22] to increase the parallelism of push-relabeling method.

In summary, due to the irregular computation and low parallelism of GC, the GC technique is not suitable for accelerating by GPU programming and VLSI design even through it can achieve more accurate disparity results.

(3) Belief Propagation

The first literature which applied to the BP approach to solve the disparity estimation problem was proposed by *Sun et al.* [24] to derive more accurate disparity maps via optimizing the energy in the graph model as shown in Fig. 2. 11. In this figure, a node represents a pixel and all nodes are connected by four-connection grid. During the optimization process, the matching costs of each node are diffused through

the messages to neighboring nodes iteration by iteration and this diffusion mechanism is called message passing. Afterward, the disparity results are determined by aggregating the matching costs and messages of a node after several iterations.

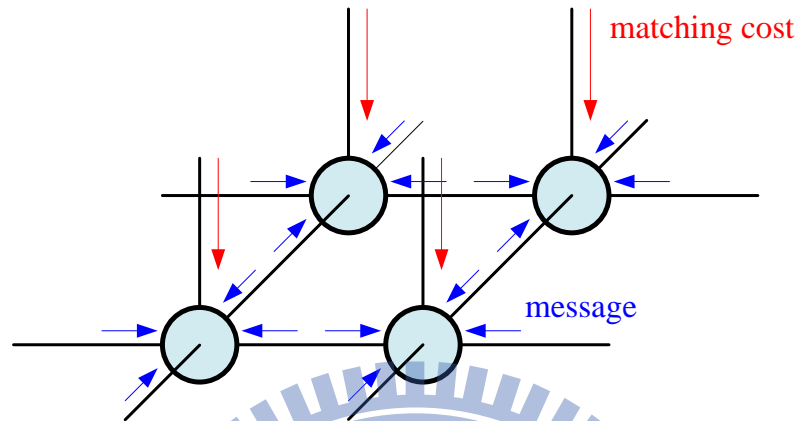


Fig. 2. 12 Illustration of belief propagation approach

The most critical issue in the BP approach is that the highest computational complexity, $O(H \times W \times DR^2 \times T)$ due to the message passing. Here, the T refers to the iteration counts. For the operation, the DR^2 results from the convolution and the iteration count T should be more than 10. Therefore, the literatures of *Felzenswalb* and *Huttenlocher* [25] proposed the hierarchical BP (HBP) and the linear-time message passing to reduce the computation of message passing. The HBP could increase the speed of disparity convergence and the linear-time message passing could reduce the complexity of convolution from $O(DR^2)$ to $O(DR)$. *Szeliski* et al. [26] proposed the max-product loopy belief propagation (BP-M) to reduce the iteration counts by a scale. However, since the BP approach has the property of high parallelism, the BP technique is much suitable to be accelerated by the GPU programming and VLSI design [27]-[33]. Unfortunately, the high memory cost ($4HW \times DR$) for storing the matching costs and messages of entire frame is the main hardware design issue. To solve this problem, the literatures of bipartite grid [25] and

the sliding approach [34] were proposed for lighting the memory access penalty and the predictive coding scheme [35] could be applied for message compression.

In summary, the DP algorithm could be easy to achieve real-time processing but suffer from the problem of streak artifacts. However, the other improvement methods would additional result in irregular computation. For the 2-D optimization techniques, although the GC technique can derive high accurate disparity map, but the irregular computational process significantly limits the capability of hardware accelerating.

Fig. 4.4. Disparity Refinement

In the final step, the post processing methods such as occlusion handling, object consistency enhancement, and temporal consistency enhancement are usually applied to further refine the disparity maps. Therefore, these methods are briefly described as follows.

(1) Occlusion Handling

The occlusion problem is defined as that the object point is visible in one view and invisible in the other view. Therefore, in the occlusion region, there are no correspondence pixels in the invisible view. In general, the incorrect disparities would appear in the occlusion regions and further induce artifacts in the view synthesis. To deal with the occlusion problem, the occlusion detection is adopted first to detect the occlusion and the occlusion filling mechanism is applied to fill the occlusion area by the background disparities in general. The basic methods for occlusion detection are surveyed in [45] based on different assumptions. The left-right check (LRC) assumes that a correspondence pair should have same disparity and the occlusion constraint (OCC) assumes that occlusion region in the other view would be resulted by the disparity gap of two pixels. In addition, the order of two pixels should have the correspondences with the same order in the other view as the order constraint (ORD)

assumed. In above occlusion detection techniques, the LRC is widely applied for the disparity refinement [6] and [40], and the OCC and the ORD are combined into the disparity optimization step [15] and [24] usually. With the detected occlusion pixels, the disparities in the occlusion region could be directly replaced by the reliable background disparities in occlusion filling step.

(2) Object Consistency Enhancement

In an object, the disparities are usually identical or changing smoothly. However, the textureless regions usually cause the incorrect disparities and thus affect the results of disparity maps. Therefore, the plane fitting approach [40] was usually adopted by the high-performance disparity estimation algorithms [44], [36], [37] to remove the disparity noise. In the plane fitting approach, the watershed segmentation, mean-shift clustering, or K-mean clustering is usually adopted for computing the segment information first. Based on the segment information, a new 3-D plane is constructed by the linear regression method using the disparities in a segment. Besides of the plane fitting method, the regional voting method [6] could also refine the disparity maps well. Compared to the plane fitting technique, the regional vote technique is much simpler due to the segment information is unnecessary.

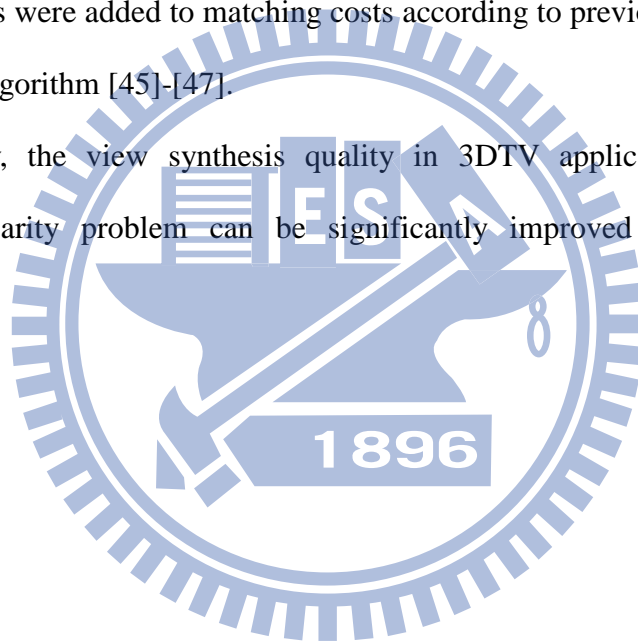
(3) Temporal Consistency Enhancement

In previous work, most disparity estimation algorithms were focused on the still image sequence [48]. However, these algorithms didn't consider the temporal consistency issue and thus result in some obstacles in the view synthesis application for video sequences. Therefore, if the temporal consistency issue has not been dealt with well, the disparity maps would produce flicker artifacts due to the independent generation of disparity for each frame. In addition, the disparities without temporal consistency treatment are unstable in the occlusion and textureless regions. As a result, the flicker artifacts would be further propagated to the view synthesis results and

becomes observed.

Intuitively, the neighboring frames are usually taken into account in the disparity estimation to address the temporal consistency issue. In the previous work [41]-[43], a disparity flow in spatial and temporal domain is constructed by buffering several disparity frames. Afterward, different smooth methods are executed in the disparity flow. On the other hand, since two adjacent frames are available, the temporal BP algorithm [38] executed the BP optimization in a 6-connection grid graph in which two additional connections are linked between previous and next frame. In addition, the temporal costs were added to matching costs according to previous disparity in the 3DVC's DERS algorithm [45]-[47].

In summary, the view synthesis quality in 3DTV applications as well as inconsistent disparity problem can be significantly improved by the disparity refinement step.



III. Proposed Dual-Way Dynamic Programming Algorithm for Disparity Estimation

3.1. Overall Flowchart

Fig. 3.1 shows the flowchart of proposed disparity estimation algorithm which is composed by three phases including Cost Calculation Phase, Disparity Estimation Phase, and Disparity Refinement Phase and the duty of each phase is described as follows in detail.

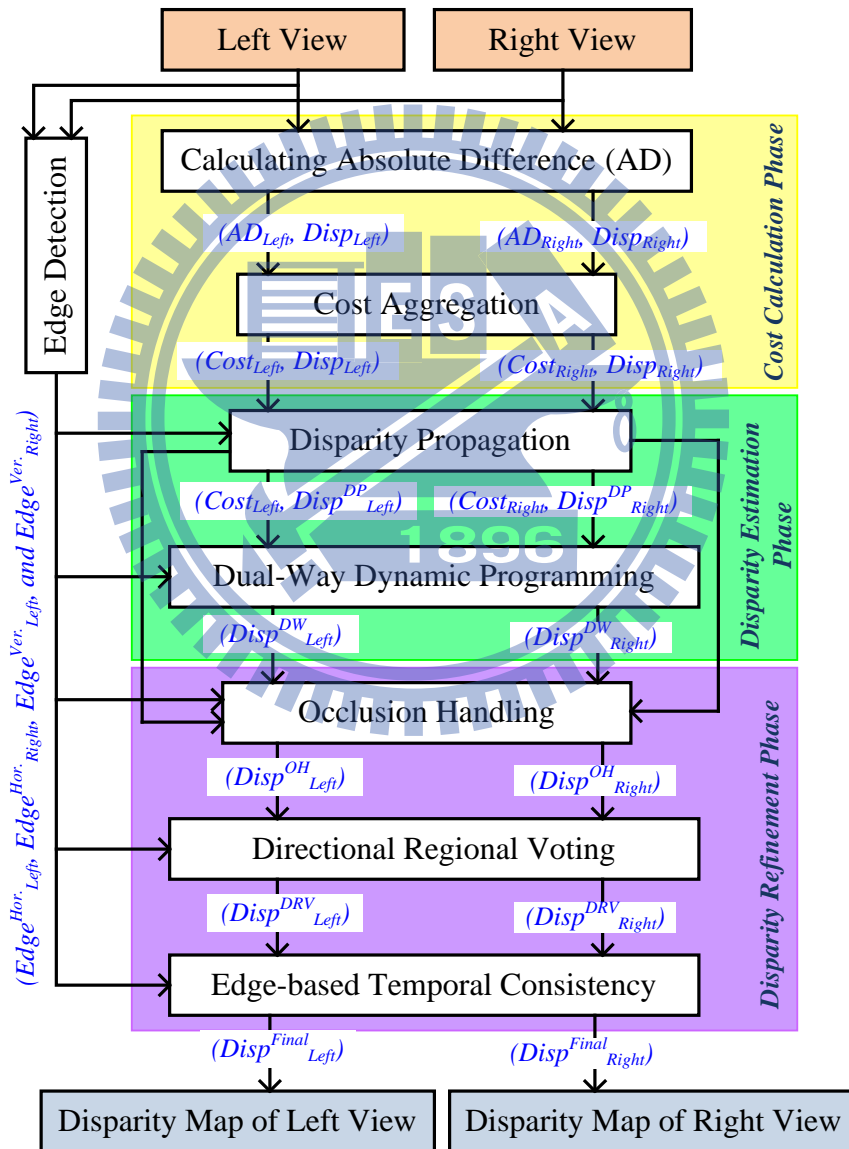


Fig. 3.1. Flowchart of proposed disparity estimation algorithm

3.2. Cost Calculation Phase

This phase corresponds to execute all required operations for deriving the costs which will be used in the upcoming disparity estimation process. In this phase, the costs of absolute difference (AD) corresponding to each pixel and disparity are calculated first. Afterward, the calculated absolute difference values will be fed into the cost aggregation module for cost refinement.

3.2.1. Pixel-based Absolute Difference

Many distortion measurements such mean square error (MSE) and sum absolute difference (SAD) have been widely adopted in many field to derive the difference between two measurement sources. In our proposed algorithm, we adopt pixel based absolute difference for deriving the distortion of each disparity for the upcoming disparity estimation process. The absolute difference of each pixel and disparity is calculated as follows.

$$AD_L(x, y, d)_{d \in DR, L \in \{Left, Right\}} = |Y(x, y) - Y'(x \pm d, y)|$$

where AD is the absolute difference; x and y stand for the pixel position; Y refers to the pixel intensity of target view; Y' means the pixel intensity of reference view, d implies the disparity, and DR represents the disparity range. Fig. 3.2 shows an example to illustrate how the absolute difference of each pixel and disparity been calculated.

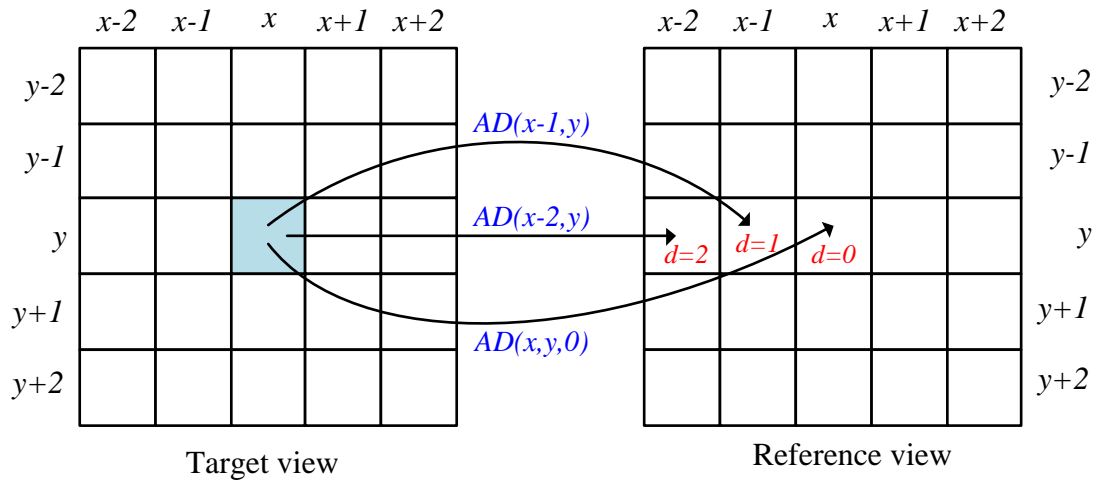


Fig. 3.2. Illustration of absolute difference calculation

3.2.2. Cost Aggregation

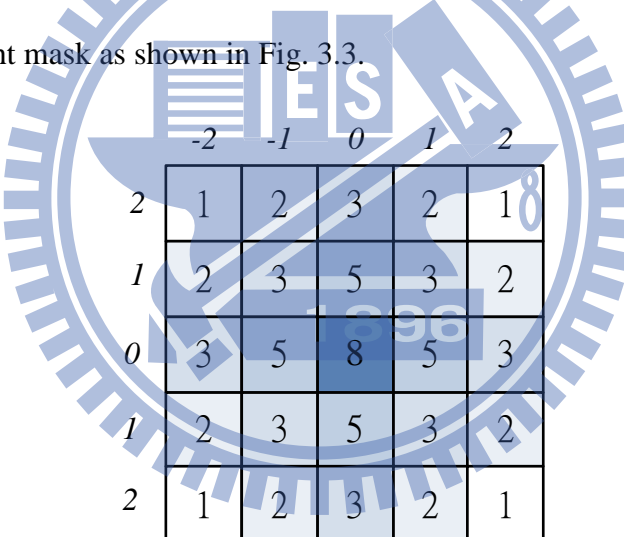
After the absolute difference calculation process, the calculated ADs will be used in the cost aggregation process for the cost refinement in each disparity level. As mentioned in Section 2 that many weighing methods can be adopted to calculate the costs for cost aggregation. However, every weighing method has its own characteristic and most suitable cases. For the simplest uniform weighting and uniform weighting with adaptive radius methods, although these methods have much simple weighting operation, they don't consider the spatial relationship and thus may result in ill aggregation results. An alternative way is the adaptive support weight method which takes the spatial relationship into account to decide the weighting coefficients. However, although adaptive support weight method can obtain much better results than uniform weighting methods, the computational complexity of adaptive weight method is significant due to the exponential operation has to be implemented. To well fit the object shape, the polygon weighting method seems to be

the good choice. However, significant memory storage spaces are required for this kind of weighting methods and thus result in the high hardware implementation costs.

As a result, the Gaussian weighting method is adopted in our proposed cost aggregation algorithm since the Gaussian weighting method has the properties of the considering the spatial relationship and low hardware implementation cost due to the fixed weighting coefficients. In our proposed algorithm, the 5×5 window size with Gaussian weight mask is used for the cost aggregation. The aggregated costs are calculated as below.

$$Cost_L(x, y, d)_{d \in DR, L \in \{Left, Right\}} = \frac{\sum_{j=-2}^2 \sum_{i=-2}^2 AD_L(x + i, j + j, d) \times W_{Gaussian}(i, j)}{\sum_{j=-2}^2 \sum_{i=-2}^2 W_{Gaussian}(i, j)}$$

where the $Cost$ is the aggregated costs and the $W_{Gaussian}$ refers to the weighting factor of Gaussian Weight mask as shown in Fig. 3.3.



	-2	-1	0	1	2
2	1	2	3	2	1
1	2	3	5	3	2
0	3	5	8	5	3
1	2	3	5	3	2
2	1	2	3	2	1

Fig. 3.3. Weighting factors of Gaussian Weight mask

3.2.3. Edge Detection

To increase the accuracy of disparity estimation, the edge information of image is considered in our proposed algorithm due to the most difficult regions to be treated always occur at the object boundaries or edges. There are many edge detection or object segmentation algorithms that have been proposed in the literatures. However, the well-known edge detection algorithm called Sobel edge detection algorithm with

3×3 window size is adopted in our proposed algorithm for detecting the edge information due to its simplicity and popularity. The edge information after Sobel edge detection process is shown in below.

$$Edge_H(x,y) = \sum_{i=-1}^1 \sum_{j=1}^1 Y(x+i,y+j) \times W_{Sobel_H}(i,j)$$

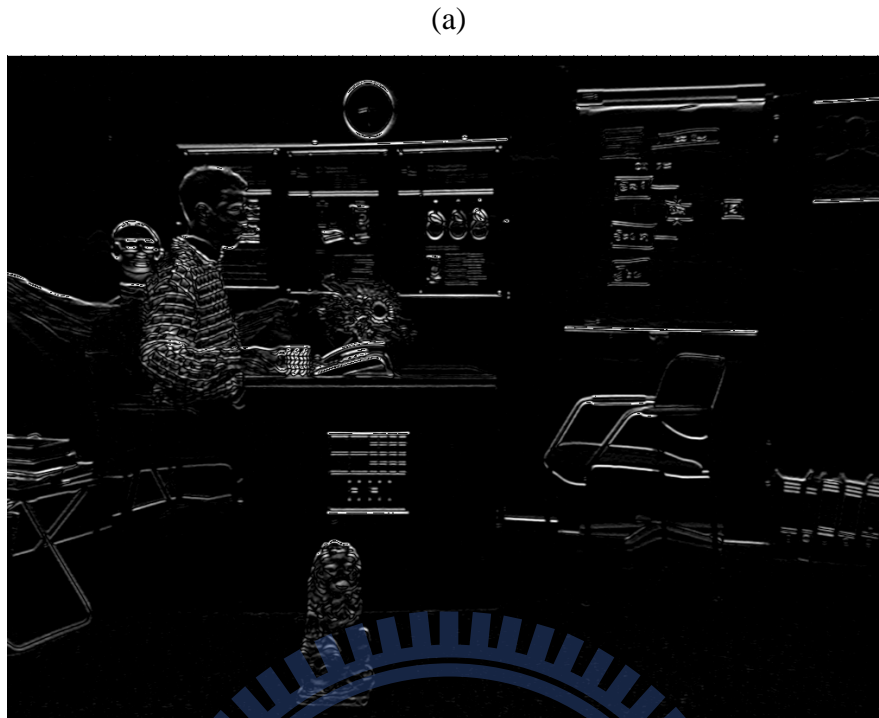
$$Edge_V(x,y) = \sum_{i=-1}^1 \sum_{j=1}^1 Y(x+i,y+j) \times W_{Sobel_V}(i,j)$$

where *Edge* is the edge information after Sobel operator; *H* and *V* refer to the horizontal and vertical direction; and W_{Sobel} means the weighting factor of Sobel mask as shown below.

$$W_{Sobel_H} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } W_{Sobel_V} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fig. 3.4 exhibits the edge information after Sobel edge detection algorithm. From this figure, we can observe that the Sobel edge detection algorithm can derive useful edge information which will be very helpful for the upcoming disparity estimation process.





(b)

Fig. 3.4. Edge information after Sobel operator (a) Horizontal edge and (b) Vertical edge

3.3. Disparity Estimation Phase

Once the costs of all pixels and disparities have been successfully calculated, the costs will be fed into our proposed disparity estimation module. In our disparity estimation phase, we propose two algorithms called disparity propagation and two-way dynamic programming as figure Fig. 3.5 shown. The disparity propagation is first executed to determine whether the neighbor disparities could be propagated to current pixel or not in order to save the computational complexity of disparity estimation. After the disparity propagation process, the proposed two-way dynamic programming algorithm is performed to obtain the disparity map of left and right view. In our proposal, the disparities are considered in both of left-to-right and right-to-left way so that the occlusion problem as well as streaking problem can be reduced efficiently. In addition, the computational complexity of our proposed disparity

estimation algorithm is much lesser than that of the traditional dynamic programming algorithms. It should be noticed that the edge detection operation is involved in our proposed algorithm since the edge information in both horizontal and vertical directions will be considered throughout the overall operation of our proposal.

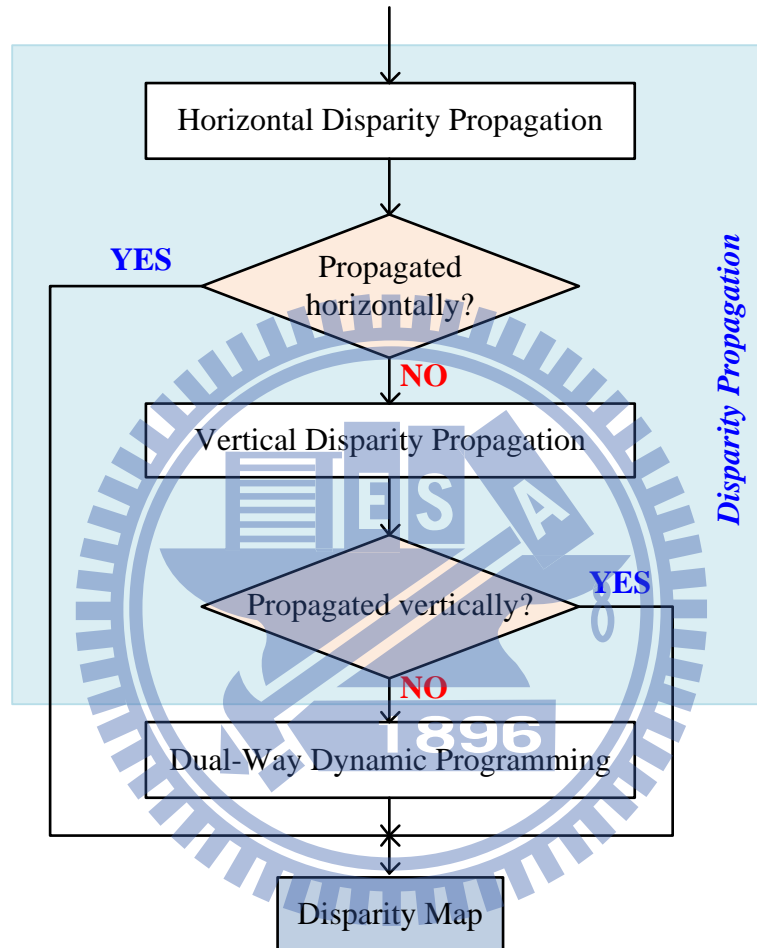


Fig. 3.5. Flowchart of proposed disparity estimation phase

3.3.1 Disparity Propagation

In the disparity estimation phase, we first use a technique called disparity propagation to determine whether to propagate the disparity from the previous pixels if the current pixel is located at the edge region. The main assumption behinds our proposed disparity propagation algorithm is that disparities around the edge area are more reliable than texture-less area, so, it is expected to propagate the disparities from

edge area to texture-less area. In addition, since the texture-less area usually contains insignificant characteristics, therefore, it is very easy to be affected by the luminance variation and thus result in the noisy results (inaccurate disparity results). Therefore, our proposed disparity propagation is tried to smooth the disparity results in horizontal and vertical direction. In addition, the disparities after disparity propagation would be as the final disparity results so that the upcoming disparity estimation process for the propagated pixels can be reduced and thus achieve the computational complexity as well as power consumption reduction. In our proposal, the disparity is propagated both from the horizontal or vertical directions and the disparity propagation operation is listed below.

Horizontal Disparity Propagation

In the horizontal disparity propagation process, both of the horizontal and vertical edge information is considered to determine the propagation of disparity. The decision rule for horizontal disparity propagation is listed below.

$$if \left(\begin{array}{l} Edge_H(x, y) \leq \delta_{Propogation} \\ Edge_V(x, y) \leq \delta_{Propogation} \end{array} \right) then D(x, y) = \underset{d \in DR}{argmin} \{Cost(x_p, y, d)\}$$

where $\delta_{Propogation}$ is the determination parameter; D refers to the final disparity map of pixel located at x and y position; and x_p means the previous pixel depends on scanning direction. Fig. 3.6 shows an example to illustrate how to derive the x_p for different scanning directions.

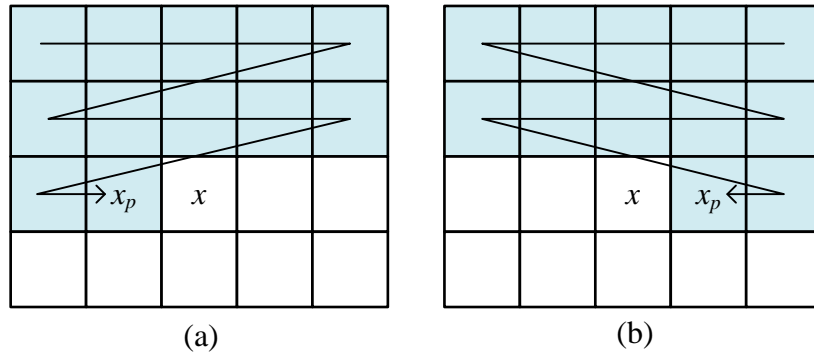


Fig. 3.6. Illustration of x_p derivation for different scanning directions

(a)Left-to-Right and (b)Right-to-Left

It should be mentioned that since our proposed horizontal disparity propagation uses two scanning directions for disparity propagation, the final disparity and cost for current pixel is decided by seeing which scanning direction has the smallest cost.

Vertical Disparity Propagation

After the horizontal disparity propagation, the vertical disparity propagation is executed depending on whether the disparity of current pixel has been propagated from the previous pixel x_p . If the disparity of current pixel is exactly propagated from the previous pixel x_p , the vertical disparity propagation will not be executed for the current pixel. Oppositely, if the disparity of current pixel is not propagated from the previous pixel x_p , the vertical disparity propagation will be on for disparity propagation. Similar to horizontal disparity propagation, the proposed vertical disparity propagation also considers the edge information from the vertical direction to decide whether the disparity could be propagated vertically. However, the main difference between horizontal disparity propagation and vertical disparity propagation is that the absolute differences of pixel intensity are further considered to decide the disparity propagation. The decision rule of our proposed vertical disparity propagation is listed below.

$$\text{if } \begin{pmatrix} Edge_H(x, y) \leq \delta_{Propogation} \\ Edge_H(x, y - 1) \leq \delta_{Propogation} \\ |Y(x, y) - Y(x, y - 1)| \end{pmatrix} \text{ then } D(x, y) = D(x, y - 1)$$

3.3.2 Dual-Way Dynamic Programming

Traditional DP usually stores all path information as a path table for path searching. Therefore, the memory storage space will be very significant due to $W(\text{frame width}) * DR$ disparities have to be stored. As a result, the memory space requirement of traditional DP is significant. In addition, the traditional DP algorithms only consider one way scanning. If only considering one way, the resulted disparity will be very bad due to the occlusion problem. Therefore, we consider both ways for improving the accuracy of derived disparity results. According to the horizontal and vertical disparity propagation result, our proposed dual-way dynamic programming algorithm is executed depending on whether the disparity of current pixel has been propagated either from the horizontal or vertical direction. If the disparity of current pixel is propagated from neither horizontal nor vertical direction, the dual-way dynamic programming will be executed to decide the final disparity of current pixel. Fig. 3.7 illustrates the idea of our proposed dual-way dynamic programming. In our proposal, the main idea is that disparities with minimum cost are considered horizontally and vertically.

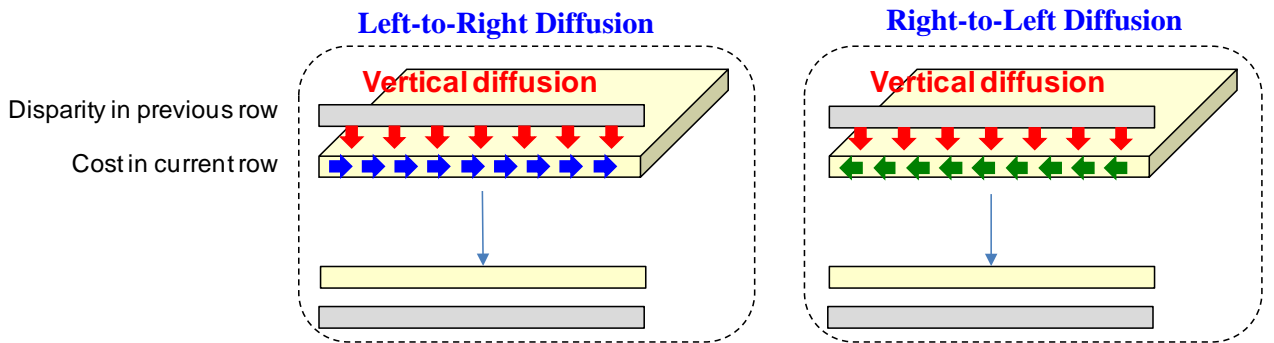


Fig. 3.7. Illustration of our proposed dual-way dynamic programming

To decide the best disparity for current pixel, we propose an energy function for deciding the disparity. The proposed energy function is listed below.

$$E(x, y, d) = \varepsilon_p \times Cost(x_p, y, d) + \varepsilon_c \times Cost(x, y, d) \\ + \rho_H \times |d'_{min} - d| + \rho_V \times |d_u - d|$$

where

ε_p : the cost penalty of previous pixel at disparity d

ε_c : the cost penalty of current pixel at disparity d

ρ_H : the horizontal smoothing cost

ρ_V : the vertical smoothing cost

d'_{min} : the disparity of previous pixel with minimum energy cost

d_u : the disparity value of pixel in upper row

Once the energy cost of each disparity level has been successfully calculated, the final disparity of current pixel can be decided by the following expression.

$$D(x, y) = \underset{d \in DR}{\operatorname{argmin}}\{E(x, y, d)\}$$

In the following, the definition of the parameters listed above is described below in detail. The main idea behinds the parameter definition of our proposed algorithm is that we pay more attention on the edge regions since the edge regions are the most difficult areas to deal with. Therefore, our parameters are defined based on horizontal and vertical edge.

For horizontal edge: Two types of horizontal edge conditions are used to decide the parameters as listed below.

1. No horizontal edge: If current pixel has no horizontal edge passed through, it

means that the smoother weighting parameters can be used for smoothing the disparity from the neighbor pixels. As a result, the parameters for no horizontal edge condition are shown in Fig. 3.8(a). This case means that the current pixel is located inside the object, therefore, we hope that the disparity of current pixel should be as much similar to the previous pixel as possible so that the resulted disparity map for certain object could be more consistent especially for the texture-less areas. Therefore, we give the higher weighting for the ε_p and ρ_h to make sure that the previous reliable disparity could affect the results of energy function significantly.

2. Horizontal edge existence: The main idea behinds this filtering is to let the disparity of current pixel to be more similar to the disparity of the object where the current pixel belonging to. If current pixel is located on the horizontal edge, two parameter definition methods are proposed depending on which region that the current pixel belongs to. If current pixel and previous pixel belong to the same object and existing the edge magnitude discontinuity in horizontal direction, the ε_p and ρ_h are increased and ε_c is decreased to result in the consistent disparity map with the streaking problem reduction as Fig. 3.8(b) shown. Otherwise, if the current pixel belongs to the right object, both of the parameter weighting of previous pixel and horizontal smooth cost are reduced. If current pixel and previous pixel belong to different objects and existing the edge magnitude discontinuity in horizontal direction as Fig. 3.8(c) shown, the ε_p and ρ_h are decreased and ε_c is increased to make sure the current disparity would be affected by the previous disparity slightly.

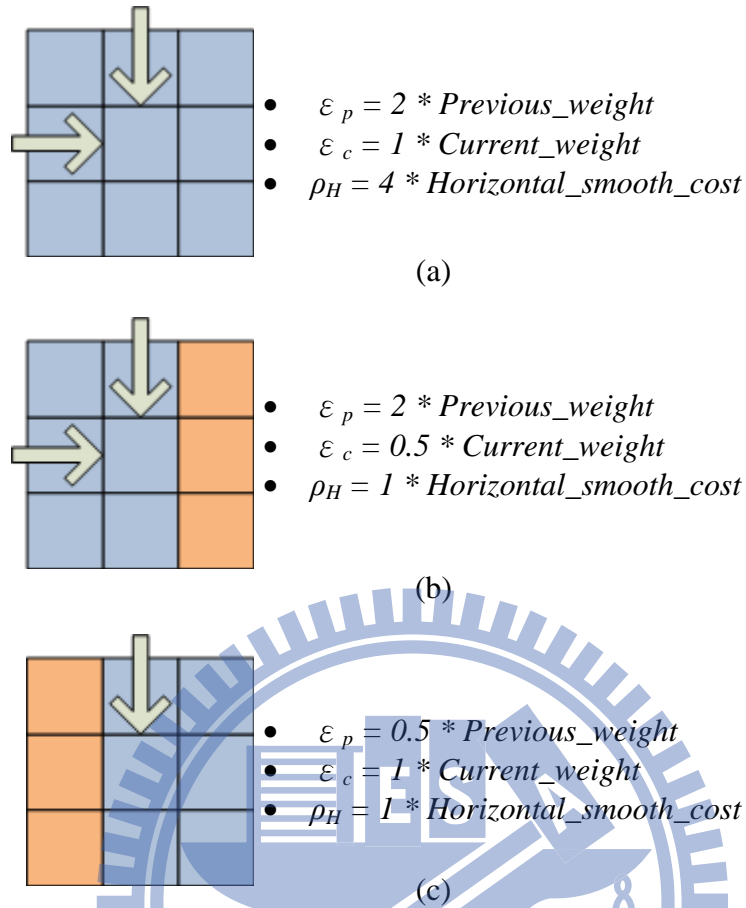


Fig. 3.8. The parameter definition for different types of horizontal edge conditions

For vertical edge: Two types of vertical edge conditions are proposed as well to decide the parameters as listed below.

1. No vertical edge: If current pixel has no vertical edge passed through, it means that the smoother weighting parameters can be used for smoothing the disparity from the neighbor pixels. As a result, the parameters for no vertical edge condition are shown in Fig. 3.9(a). This case means that the current pixel is located inside the object, therefore, we hope that the disparity of current pixel should be as much similar to the previous pixel as possible so that the resulted disparity map for certain object could be more consistent especially for the texture-less areas.

Therefore, we give the higher weighting for the ρ_v to make sure that the previous reliable disparity could affect the results of energy function significantly.

2. Vertical edge existence: If current pixel is located on the vertical edge, two parameter definition methods are proposed depending on which region that the current pixel belongs to. If current pixel and upper pixel belong to the same object and existing the edge magnitude discontinuity in vertical direction, the ρ_v is decreased to result in the consistent disparity map with the streaking problem reduction as Fig. 3.9 (b) shown. Otherwise, if current pixel and upper pixel belong to different objects and existing the edge magnitude discontinuity in vertical direction as Fig. 3.9(c) shown, the ρ_h is decreased to make sure the current disparity would be affected by the previous disparity slightly.

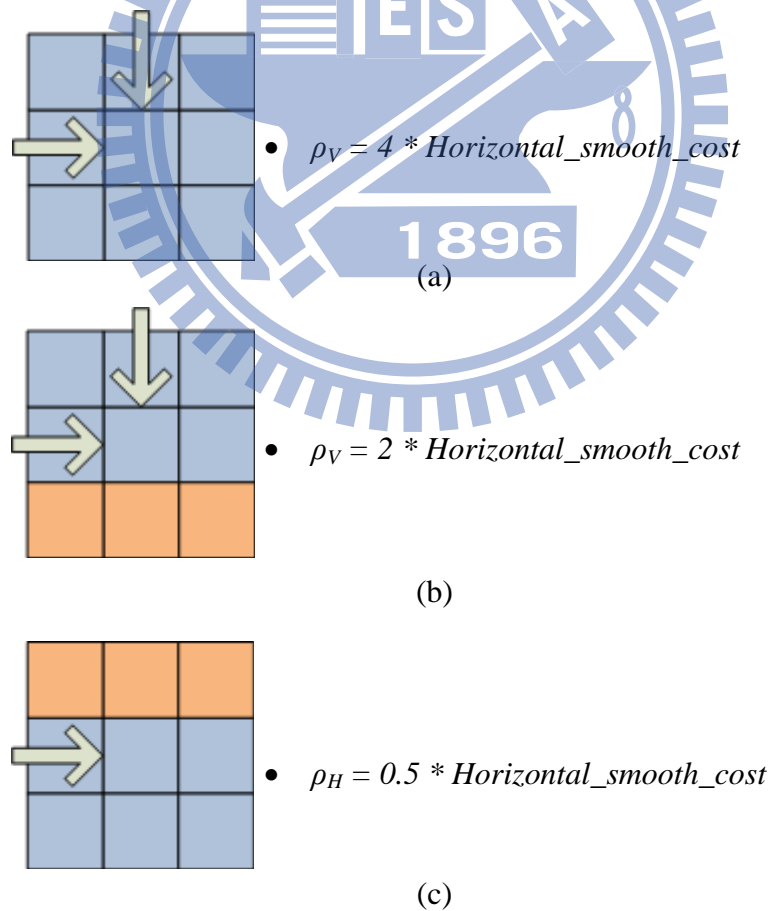


Fig. 3.9. The parameter definition for different types of vertical edge conditions

In our thesis, the constant parameters of *Previous_weight*, *Current_weight*, *Horizontal_smooth_cost*, and *Vertical_smooth_cost* are set to 1, 2, 0, and 16, respectively by empirical approach.

Fig. 3.10 shows the disparity estimation results of our proposed algorithm. From this figure, we can observe that the disparity propagation can result in much more smooth disparity estimation result compared to only using dual-way dynamic programming disparity estimation.





(b)

Fig. 3.10. Results of disparity estimation after (a) dual-way dynamic programming and (b) dual-way dynamic programming with disparity propagation

3.4. Disparity Refinement Phase

Although the disparity estimation phase can derive the disparity maps for both of left and right views, however, the disparity maps should be further refined to obtain more accurate disparity estimation results. Therefore, three algorithms called occlusion handling, directional regional voting, and edge-based temporal consistency are proposed in our disparity refinement phase to further punish the disparity maps. In occlusion handling, the occlusion artifacts are treated in order to reduce the artifact effects in the occlusion regions. The directional regional voting process performs the filtering operation for the disparity map in a directional regional manner so that the processed disparity map could be as smoother as possible. Finally, the edge-based temporal consistency operation executes the filter operation for the disparity map in the temporal domain by considering not only the color difference but the edge

information so that the filtered disparity map could be very smooth between the consecutive frames. The detailed design principle of each proposed module will be described in the following subsections.

3.4.1 Edge-Based Occlusion Handling

In general, the occlusion effects usually occur at the object edge and boundary. Therefore, occlusion handling by considering the edge information is a very intuitive and straightforward manner. In our proposal, the idea of strong edge is detected to help the occlusion handling. Fig. 3.10 shows the illustration and flowchart of our proposed occlusion handling algorithm. Our proposed occlusion handling algorithm is composed by three steps called *Strong Edge Detection*, *Reliability Check* and *Section Voting and Smoothing*.

Strong Edge Detection: In strong edge detection module, the edge information after edge detection is used to determine the strong edge positions by the following procedure. The variables inside the pseudo code are defined as follows.

POS_{Left} : Position to indicate the location of left strong edge

POS_{Right} : Position to indicate the location of right strong edge

$Image_{Height}$: Image height

$Image_{Width}$: Image width

δ_{OC} : Threshold for strong edge detection

```

Set  $Pos_{Left}$  and  $Pos_{Right}$  to 0
FOR  $y = 0$  to  $Image_{Height}$ 
  FOR  $x = 0$  to  $Image_{Width}$ 
    IF ( $Pos_{Left}$  has been found)
      IF(  $|Edge_H(x,y)| + |Edge_V(x,y)| \geq \delta_{OC}$  )
         $Pos_{Right} = x;$ 
      End IF
      IF (Both  $Pos_{Left}$  and  $Pos_{Right}$  have been found)
        Reliability Check();
        Section Voting and Smoothing ();
         $Pos_{Left} = Pos_{Right};$ 
      End IF
    Else
      IF(  $|Edge_H(x,y)| + |Edge_V(x,y)| \geq \delta_{OC}$  )
         $Pos_{Left} = x;$ 
      End IF
    End IF
  End FOR
End FOR
End FOR

```

Reliability Check: Once two strong edge positions Pos_{Left} and Pos_{Right} have been successfully detected, the disparities located between Pos_{Left} and Pos_{Right} are checked one by one to determine whether the disparity is reliable enough or not. If a disparity has been determined as unreliable, it will be marked as unreliable one and it will not be considered in the following disparity smoothing operation anymore. The idea behinds our reliability checking is to avoid the disparities that have been influenced by the unreliable disparities inside a restricted region. Determining whether a disparity is reliable or not can be achieved by the following operation.

$$if \begin{cases} |D(x, y) - D(x - D(x, y), y)| \leq \delta_d & D(x, y) \text{ marks as reliable} \\ \text{Otherwise} & D(x, y) \text{ marks as unreliable} \end{cases}$$

where $x \in \{Pos_{Left}, Pos_{Right}\}$ and δ_d is an empirical defined threshold.

Section Voting and Smoothing: After the unreliable disparities have been marked, the section voting and smoothing operation is executed for smoothing the disparities located between Pos_{Left} and Pos_{Right} . Fig. 3.11 exhibits a step-by-step illustration for explaining our proposed section voting and smoothing algorithm. First, the disparity and reliable/unreliable maps are fed into the section voting operation for counting the occurrences of each disparity with the help of reliable/unreliable map. If a disparity has been marked as unreliable, the counts of corresponding disparity will not be accumulated. Once the section voting operation has been done, the disparity with maximum occurrence will be selected as the dominating disparity and this disparity will be used to replace all disparities located between Pos_{Left} and Pos_{Right} .

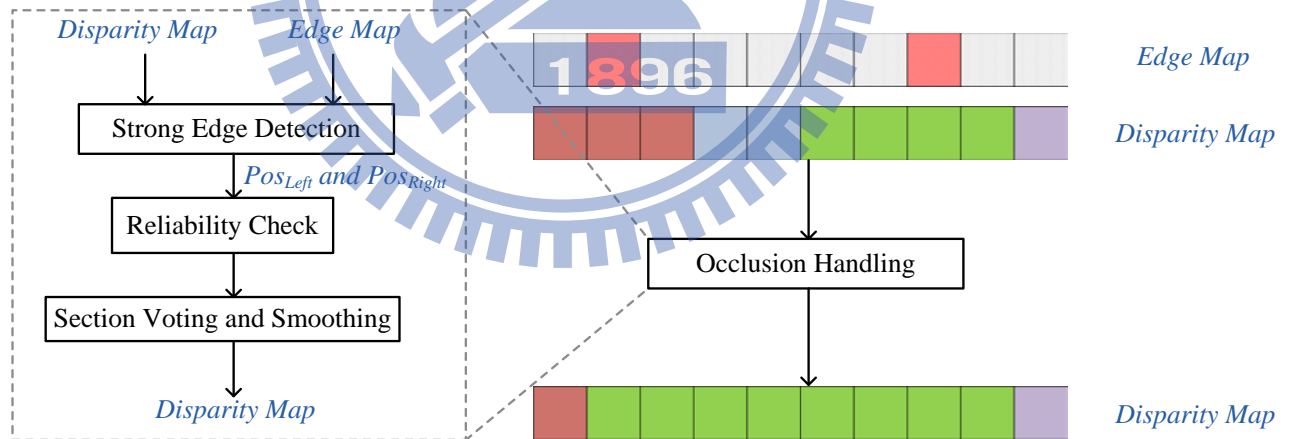


Fig. 3.11. Illustration and flowchart of our proposed occlusion handling algorithm

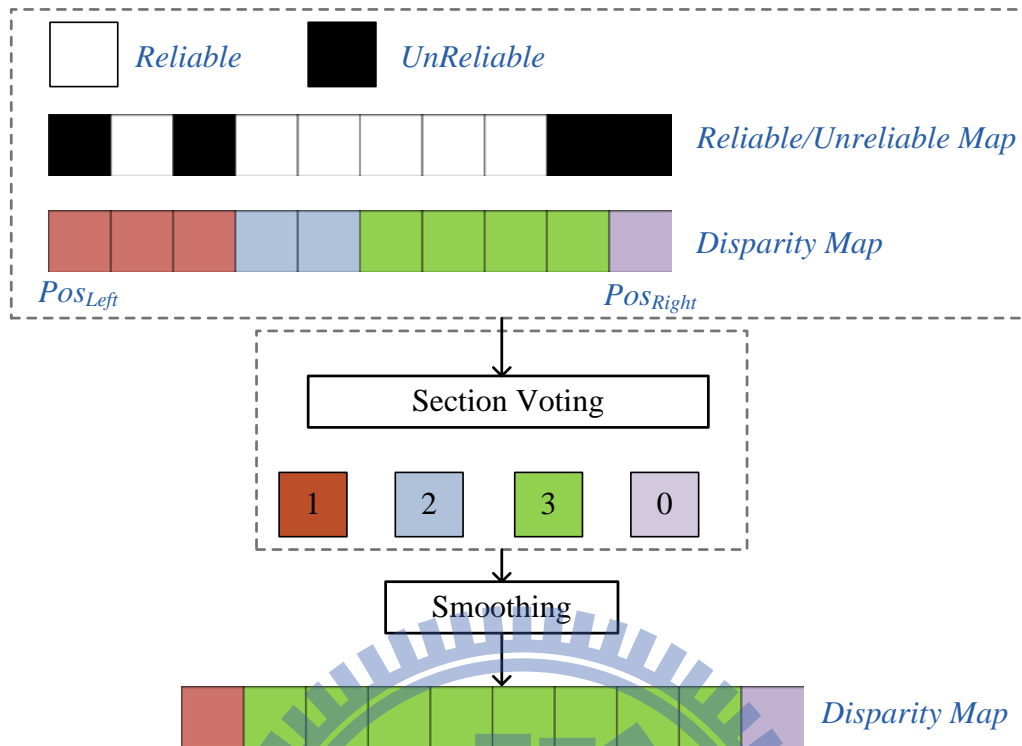


Fig. 3.12. Illustration of our proposed Section Voting and Smoothing algorithm

3.4.2 Directional Region Voting

In the traditional region voting approaches, a voting window is used for voting the occurrence of each disparity. However, this voting approach suffers from the inharmonious disparity effect inside the voting window if the covered region of voting window contains two objects. In this situation, the traditional region voting approaches would further damage the estimated disparity results. Therefore, we propose a directional region voting algorithm to avoid the problem caused by the traditional region voting via considering both of the edge and color difference information. Following figure is the diagram of procedure of proposed directional regional voting method.

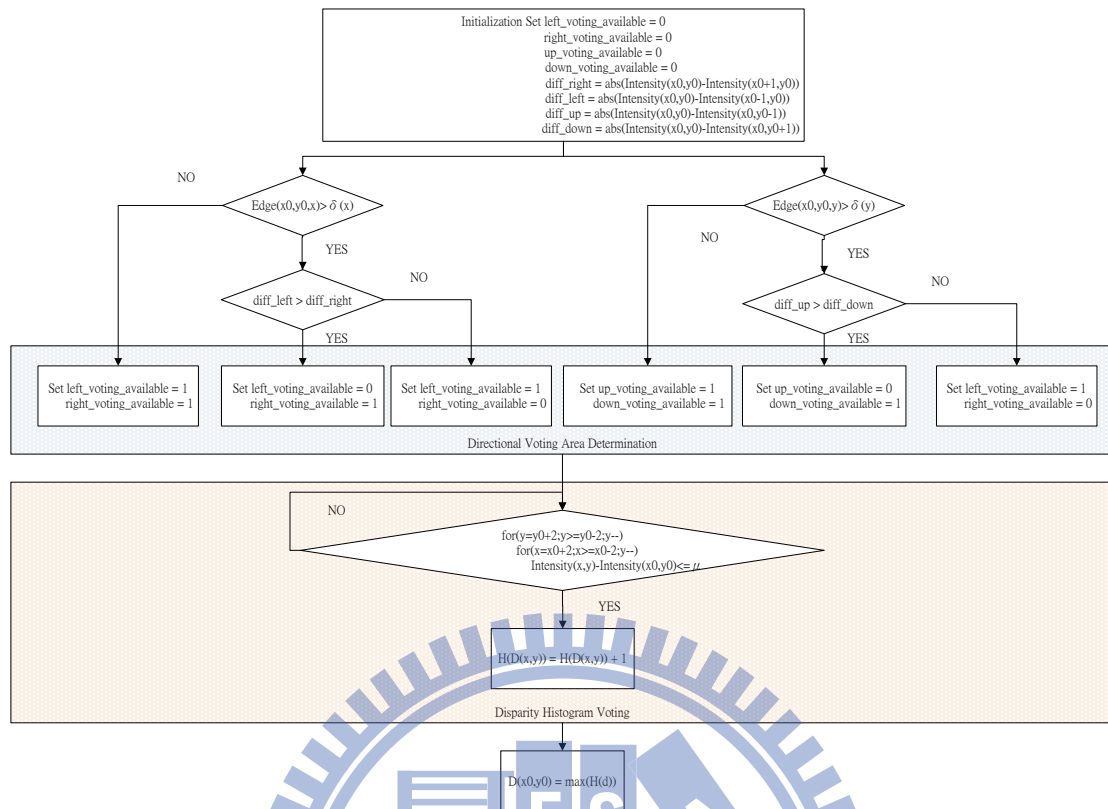


Fig. 3.13. Flowchart of proposed directional regional voting algorithm

According to the algorithm, there are nine types of voting area that will be possible to be determined. Please check them as figure below:

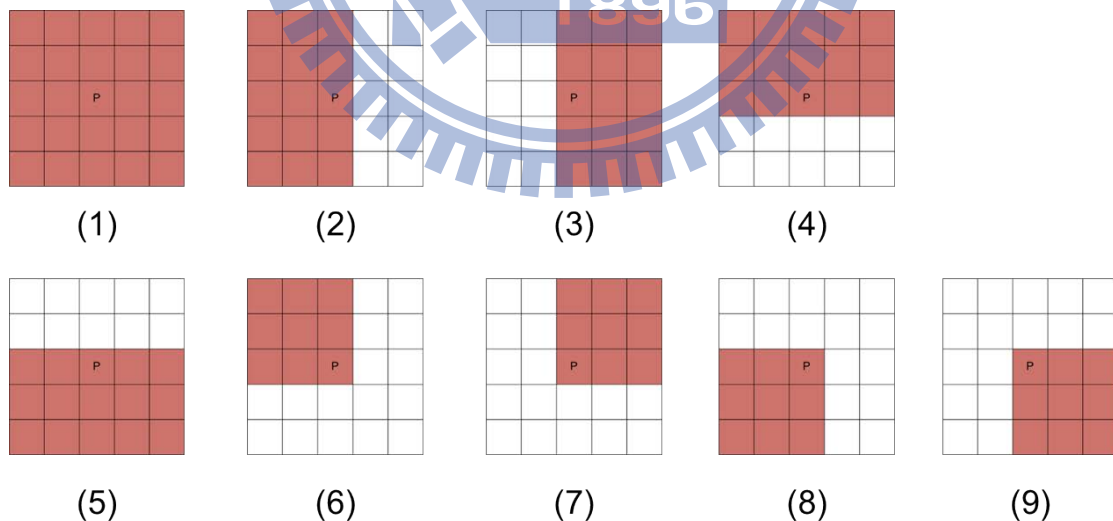


Fig. 4.14. nine types of determined voting area

Figure 3.14 shows nine different types of determined area which might occur in the algorithm. The determined voting area is determined by the edge information on

both horizontal and vertical direction and also take the intensity difference between center pixel and other neighbor pixels into account. After the voting area is determined, the algorithm will start accumulating disparity histogram of the pixels in the determined region. Only the ones which have similar intensity value will be taken into account. After the disparity histogram is complete, the disparity which get the maximum number will be assigned to the center pixel.

3.4.3 Edge-based Temporal Consistency

In traditional temporal consistency treatment, the color difference between successive frames is used for determining whether the disparity of current frame should be replaced by the disparity from the collocated pixel in previous frame to keep the disparity consistence temporally. However, as mentioned before that the occlusion effect usually occurred at the edge region. Therefore, our proposed algorithm will not only consider the color difference but also the edge information for the temporal consistence treatment. In addition, three consecutive frames are considered at the same time for deciding the disparity propagation.

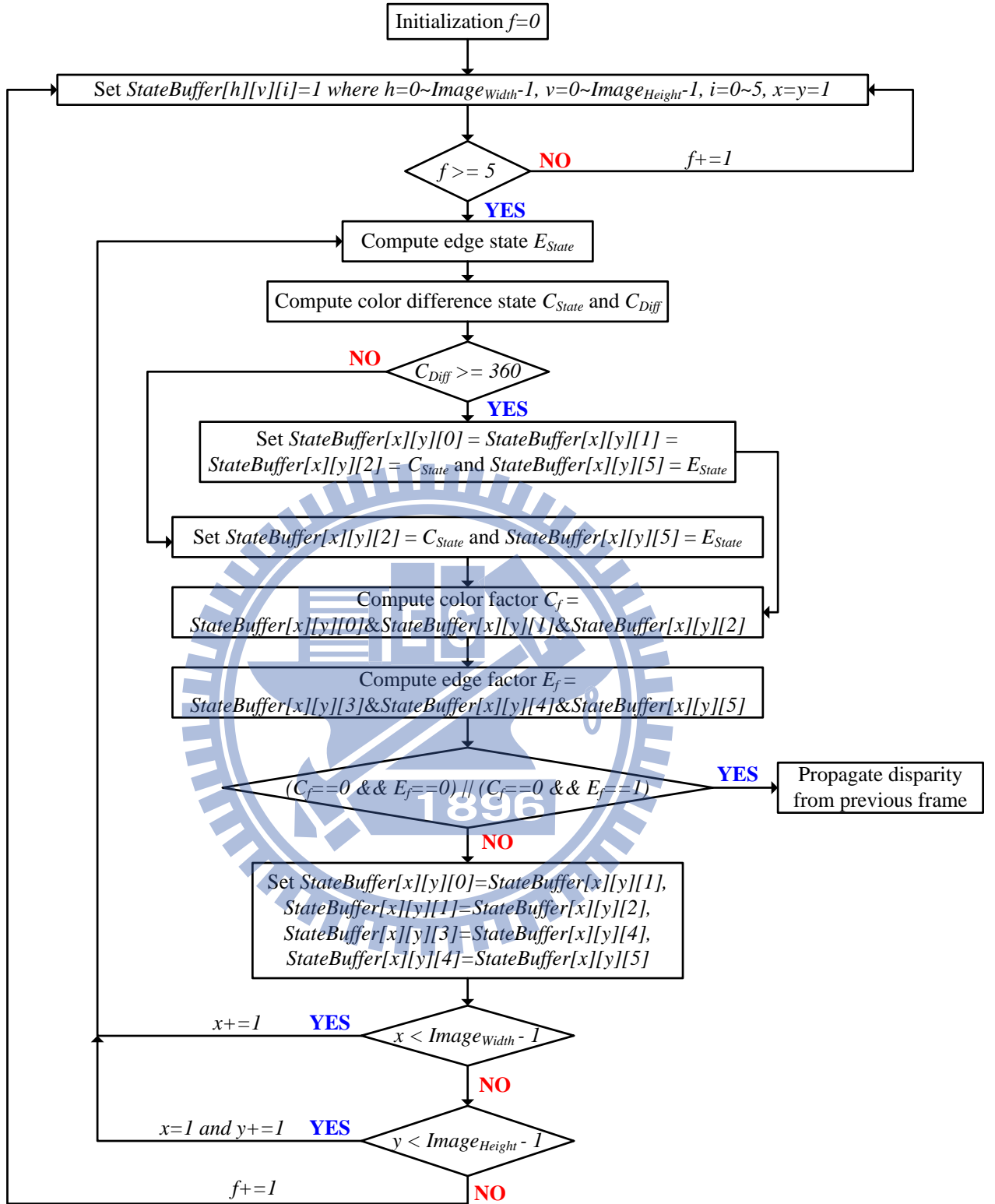


Fig. 4.15. Flowchart of proposed temporal consistency algorithm

3.5. Simulation Results

Our proposed algorithm has been tested on some general test contents, Fig. 3.16 shows the contents adopted in the experiment, and table III-1 lists their detailed information. The test sequences are provided by different research institutes. The frame size includes 1024×768 (XGA), 1920×1080 (HD1080p), and 1280×960. In these sequences, the Kendo, Balloons, Hall1, and Hall2 are captured by the moving cameras, and others are captured by fixed cameras. All the test sequences have been calibrated to make sure the disparity range can be limited in 1-D space.



Fig. 4.16. Clips of test sequences in center view

(a) BookArrival, (b) LoveBird1, (c) Newspaper, (d) Café, (e) Kendo, (f) Balloons, (g) Champagne, (h) Pantomime, (i) Hall1, (j) Hall2, (k) Street, (l) CarPark

Table III-1 Test sequences

Sequence Name	Provider	Frame Size	Frame Rate (frame/s)	Number of Frame	Number of View	Camera Spacing (cm)	Is Moving Camera
BookArrival	HHI	1024×768	16.67	300	16	6.5	No
LoveBird1	ETRI	1024×768	30	300	12	3.5	No
Newspaper	GIST	1024×768	30	300	9	6.5	No
Kendo	Nagoya	1024×768	30	300	7	5	Yes
Balloons	Nagoya	1024×768	30	300	7	5	Yes
Champagne	Nagoya	1280×960	30	300	80	5	No
Pantomime	Nagoya	1280×960	30	300	80	5	No

HHI: Fraunhofer Heinrich Hertz Institute, Germany

ETRI: Electronics and Telecommunications Research Institute, Korea

GIST: Gwangju Institute of Science and Technology, Korea

Nagaya: Nagoya University, Japan

1.1. Organization

The rests of this thesis is organized as follows. In Section II, some related works including disparity estimation, general algorithm flow are overviewed to establish the related background of the main target that this thesis would like to address. Section III presents the detailed algorithm description that this thesis proposed. Some simulation results compared to other literatures are given in Section VI to show the efficiency of our proposal. The hardware architecture design for our proposed algorithm is discussed in Section V. Finally, some conclusions and future works are given in Section VI.

3.5.1 Input and Output Configuration

The input sequences are in 2-view configuration. Table III-2 shows the selected views of all the test sequences for 2-view configuration. Table III-3 shows the output configuration and the selected views in our evaluation.

Table III-2 Input and output views for 2-view configuration

Sequence Name	Input View No.	Synthesized Pair	Frame Range for
	(I_L-I_R)	$(I_L-V_C$ or $V_C-I_R)$	Disparity Estimation
BookArrival	10-8	10-9	0-99
LoveBird1	6-8	7-8	0-299
Newspaper	4-6	5-6	0-299
Kendo	2-4	3-4	0-299
Balloons	2-4	3-4	0-299
Champagne	39-41	40-41	0-499
Pantomime	39-41	40-41	0-499

Table III-3 Experiment setting in our evaluation

Sequence Name	Output	Frame Size	Disparity	Frame	Proposed Algorithm		
	No.		Range	Range	Input No.	Avail.	Eval.
BookArrival	9, 7	1024×768	70	0-99	10-8-6	Yes	Yes
LoveBird1	5, 7	1024×768	70	0-299	4-6-8	Yes	Yes
Newspaper	3, 5	1024×768	88	0-299	2-4-6	Yes	Yes
Kendo	2, 4	1024×768	64	0-299	1-3-5	Yes	Yes
Balloons	2, 4	1024×768	64	0-299	1-3-5	Yes	Yes
Champagne	38, 40	1280×960	110	0-499	37-39-41	Yes	Yes
Pantomime	38, 40	1280×960	40	0-499	37-39-41	Yes	Yes

3.5.2 Experiment Results

The experiment settings follow the above description in previous section. We test three different kinds of evaluation statistics as following, PSNR, PSPNR, computation complexity reduction. Table III-4 shows the result of different sequences in PSNR, PSPNR, and table III-5 shows the computation complexity reduction of sequences.

Method Type Sequence	Proposed		RTHDSM[54]		HQDE[55]	
	PSNR	PSPNR	PSNR	PSPNR	PSNR	PSPNR
BookArrival	35.41	49.21	35.36	48.60	35.89	51.83
Pantomine	37.32	55.30	37.89	56.98	37.10	51.88
Lovebird1	34.16	51.37	33.83	51.89	31.86	51.92
NewsPaper	29.57	43.29	29.32	42.11	29.86	44.06
Kendo	35.94	49.31	35.58	49.64	35.66	49.90

Table III-4 Experiment Results

Method Type	Proposed
Sequence	
BookArrival	56.3%
Pantomine	66.6%
Lovebird1	35.01%
NewsPaper	41.9%
Kendo	59.8%
Champagne	57.6%
balloons	54.07%

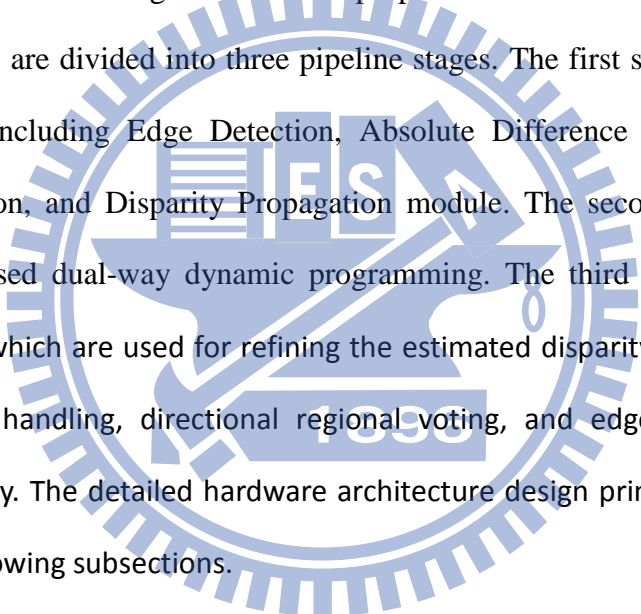
Table III-5 Computational complexity reduction

According to above experiment results, we can see that the proposed algorithm has generated good quality of disparity map by using dual way dynamic programming, edge-based occlusion handling, and directional regional voting techniques. Table III-5 also shows that the proposed algorithm solves the flickering problem effectively by using edge-based temporal consistency techniques.

IV. Hardware Implementation and Results

4.1 Overall architecture

Fig. 4.1 shows the overall hardware architecture design of our proposed disparity estimation algorithm. In our proposed hardware design, the proposed algorithms are divided into three pipeline stages. The first stage contains four modules including Edge Detection, Absolute Difference Calculation, Cost Aggregation, and Disparity Propagation module. The second stage contains our proposed dual-way dynamic programming. The third stage contains all modules which are used for refining the estimated disparity results including occlusion handling, directional regional voting, and edge-based temporal consistency. The detailed hardware architecture design principle is explained in the following subsections.



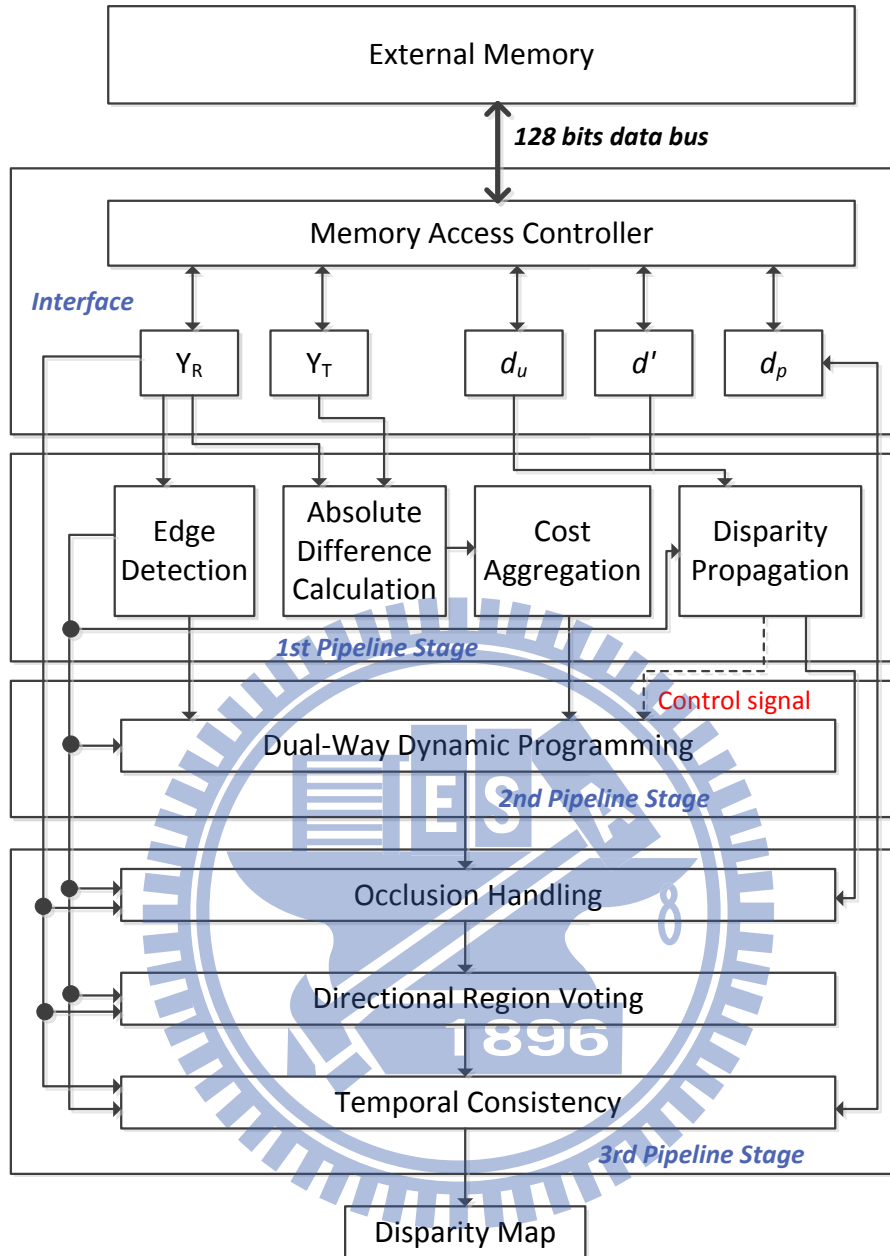


Fig. 4.1. The hardware architecture of our proposed disparity estimation algorithm

4.2 First Stage Design

- Absolute Difference Calculation Module

The operation of our proposed absolute difference calculation module is described as follows. First, three rows of luma input pixel are read from external memory and buffered by using internal memory buffer. Afterwards, 5x5 pixels are

read from internal memory left to right and fed into absolute difference computation module. In this thesis, 5x5 pixels are grouped together as the basic processing unit. For each 5x5 processing unit, the absolute differences within the disparity range are calculated for the following disparity estimation purpose. Therefore, number of 8-bit adders depends on the disparity range. Adders are required for computing all absolute differences. Fig 4.2 exhibits the detailed hardware architecture design of our proposed absolute difference calculation module. In this design, two buffers are used for storing the image data of Y_R and Y_T . In addition, the number of absolute difference calculation sets with 25 parallel absolute difference calculation modules in each is depending on the disparity range.

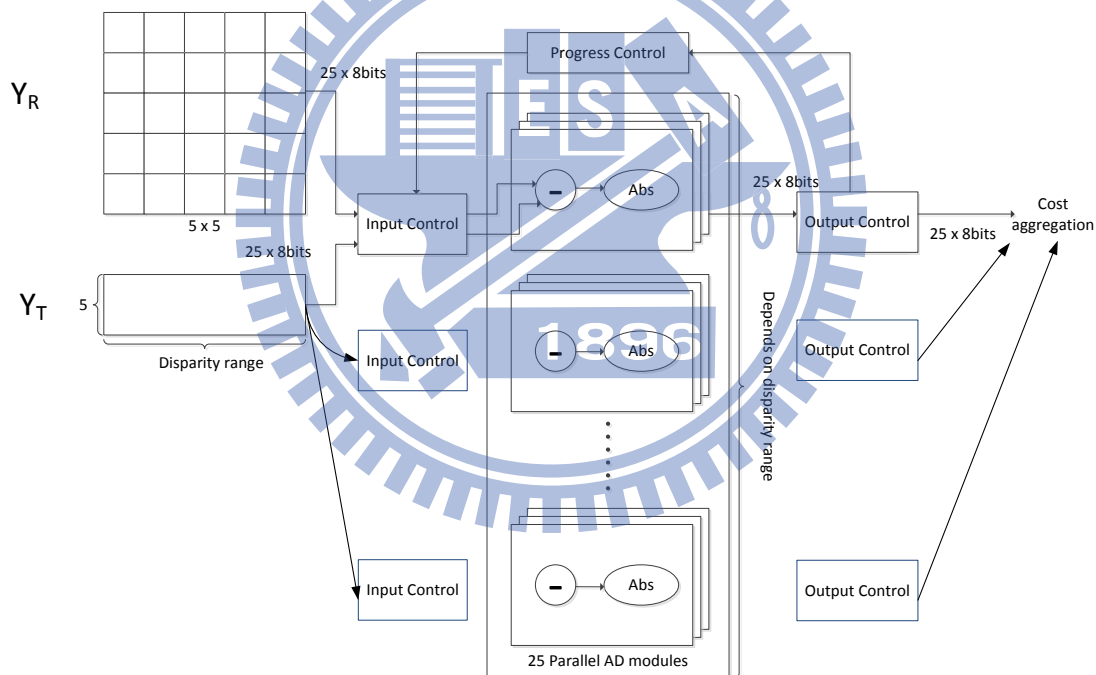


Fig. 4.2. The hardware architecture of our proposed absolute difference module

- Cost Aggregation Module

After obtaining the absolute differences, the aggregation operation is applied to the absolute differences for deriving the cost aggregation results. Although our proposed aggregation algorithm only directly implements the cost aggregation

operation as Fig 3 shown, however, it is worth to point out that our proposed hardware architecture has a very good design that the absolute differences of each 3x3 processing unit are directly added by the Gaussian weightings and sent to the disparity propagation module without buffering so that the hardware buffer costs can be reduced significantly.

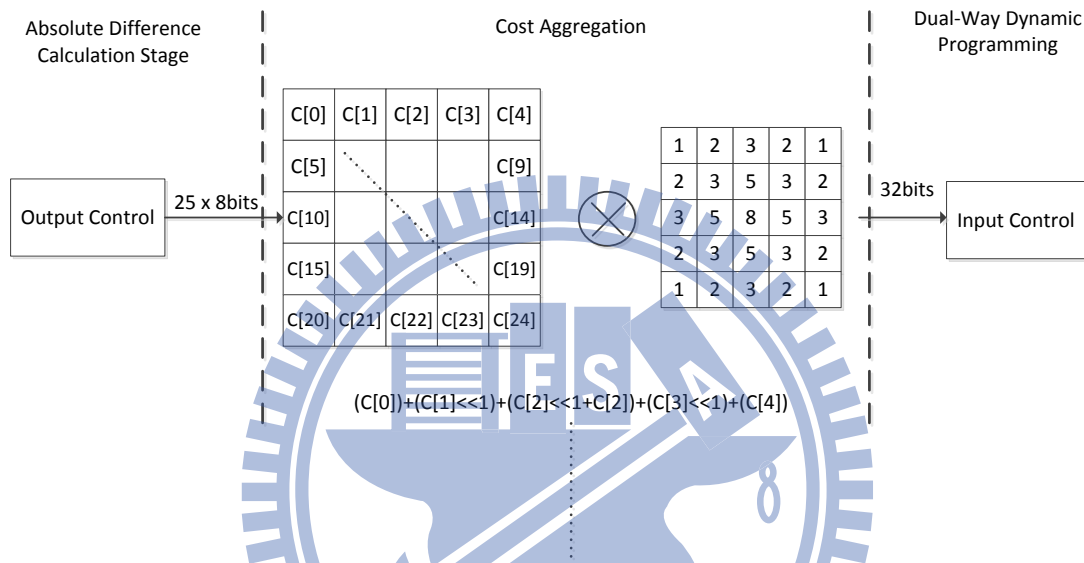


Fig. 4.3. The hardware architecture of our proposed cost aggregation module

- Disparity Propagation Module

As mentioned before that the edge information is used to determine whether the disparity of current pixel should be propagated from the disparity of previous of above pixel. Therefore, the cost propagation module must contain edge detection module to obtain the edge information of current 3x3 processing unit. Just like the absolute difference calculation module, the luma components stored inside the internal memory buffer with size in 3x3 are accessed and then processed by Sobel operator. Once the edge information has been obtained, the edge information is used to determine whether the disparity should be propagated. If the result shows

that the disparity of current pixel should be propagated, the disparity information of previous or above pixel would be read from DRAM.

4.3 Second Stage Design

When designing the hardware architecture of dynamic programming, the most critical issues is the data dependency existed in the energy function since the energy function takes the neighbor information into account for deciding the disparity of current pixel. Therefore, in order to break the data dependency and thus helping to increase the possibility of parallelization, we remove the smooth term of $(\rho_H \times |d'_{min} - d|)$ from the energy function of dynamic programming. By removing this term, the operation of each pixel will not be affected by the result of previous pixel and thus lets the different pixels belonging to the same scanline could be able to execute dynamic programming in parallel. Since the data dependency has been removed, our proposed architecture calculates the energy function of two adjacent pixels to increase the throughput so that our specification can be met. Therefore, to calculate the energy function of two pixels at the same time, the aggregation information of four pixels has to be inputted. When the costs of two pixels have been inputted to the dynamic programming module, the dynamic programming module computes the energy costs from left to right and right to left. However, thanks the removing of the horizontal term of $(\rho_H \times |d'_{min} - d|)$, it would not result in wrong results of the parallel computed energy function. Once all 128 energies corresponding to 128 disparities have been successfully calculated, the energies are compared and the two disparities with least energy cost will be stored into the buffer as the final results. Here, the hardware cost to compare the energies is 33 bits comparator x 127 x 2(one for left to right and another one for right to left).

further refining the estimated disparity map. The detailed architecture design is explained as follows.

- Occlusion Handling Module

Fig 4.6 shows the detailed hardware architecture design of our occlusion handling module. In our design, the results produced by dual-way dynamic programming stage are inputted to the occlusion handling module. The Input Control module controls all operations of all sub-modules. The Occlusion Strong Edge Detection module takes the edge information resulted from first pipeline stage to detect the strong edge. The Disparity Histogram module counts the appearance of each disparity and the results will be produced by the Disparity Assignment module.

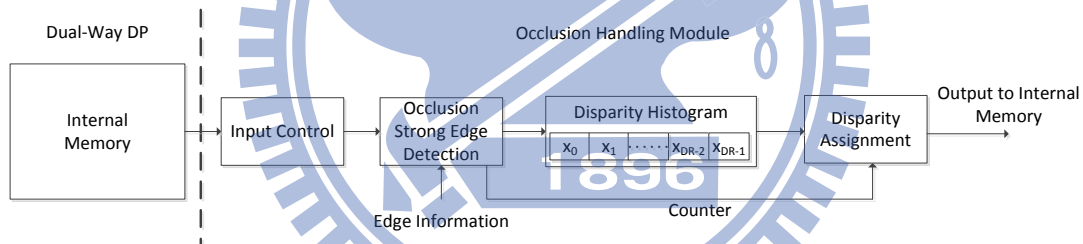


Fig. 4.6. The hardware architecture of our proposed occlusion handling module

- Directional Regional Module

Fig 4.7 shows the detailed hardware architecture design of our directional region voting. The directional voting arbiter takes 5x5 inputs from outputs of edge-based occlusion handling and also the edge information into account. The directional voting arbiter determines the available area for voting and then do the voting with disparity histogram. After the histogram is complete, we assign the disparity of maximum histogram number to the center pixel.

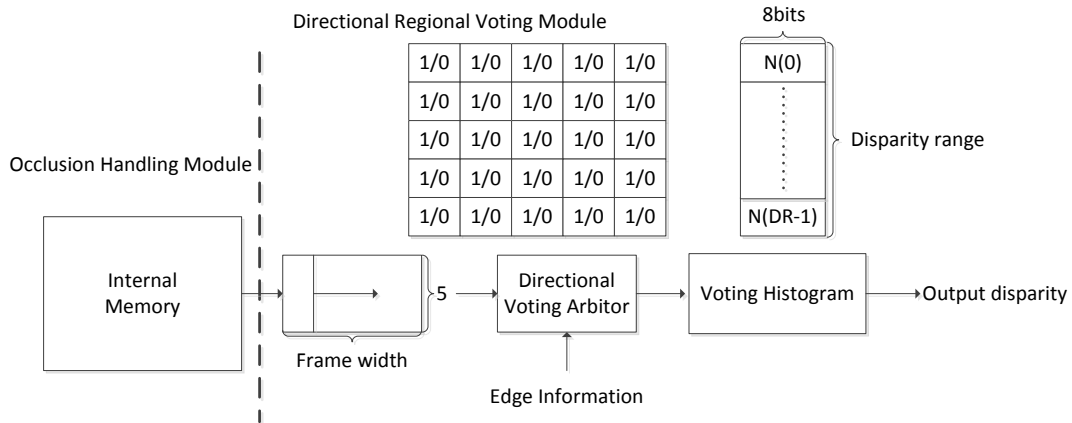


Fig. 4.7. The hardware architecture of our proposed directional region voting module

- Temporal Consistency Module

Fig 7 shows the detailed hardware architecture design of our temporal consistency module. The temporal consistency kernel takes the intensity value of current pixel, the corresponding pixel in the previous frame, and edge information into account to check if the disparity of pixel has to be propagated or not. The kernel check the difference between two corresponding pixels and also check the edge information at both horizontal and vertical directions to make sure if the current pixel is fixed or moving.

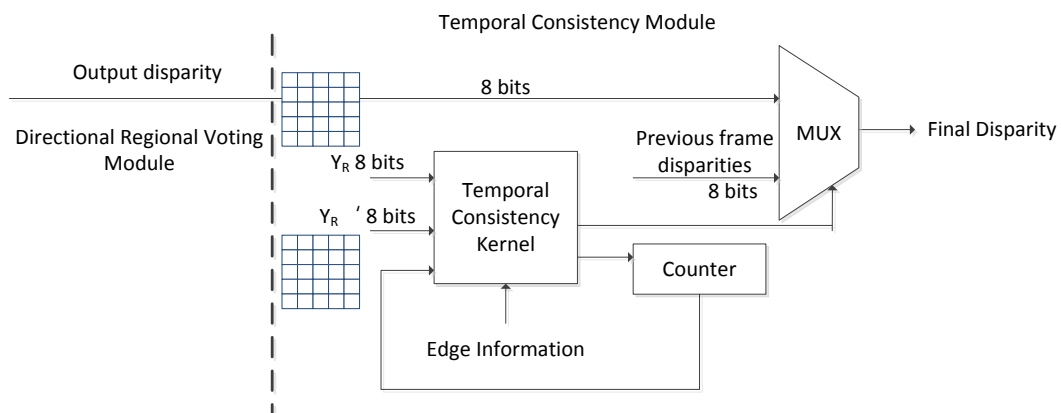


Fig. 4.8. The hardware architecture of our proposed temporal consistency module

4.5 Implementation Results

Table III-6 tabulates the hardware implementation results. The processing ability of our designed hardware architecture is processing 1080p frame with 90 fps frame rate. The disparity range is 128 so that our proposed design can search very wide range. In addition, our design is synthesized by UMC 90nm CMOS technology. The synthesized results report that our proposed hardware architecture consumes 2325K gate count and 80KB SRAM space under the 100MHz operating frequency and 128 bits bus bandwidth.

Input image resolution	1920x1080
Output image resolution	1920x1080
Frame rate	90 fps
Disparity range	128
Technology	UMC 90nm CMOS
Gate counts	2325K
Operating frequency	100MHz
Bus bandwidth	128 bits
SRAM (Internal memory)	80KB

Table III-6 Implementation results

V. Conclusion and Future Work

5.1 Conclusion

The main contribution of this thesis is to propose a low computational complexity disparity estimation algorithm with good quality. We contribute in every single stage of the algorithm such as “Dual Way Dynamic Programming”, “Disparity Propagation”, “Edge-Based Occlusion Handling”, “Directional Regional Voting”, and “Edge-based Temporal Consistency”. The algorithm yields good quality compared with other disparity estimation methods and also is hardware-friendly to be implemented. The implementation takes reasonable cost on both gate counts of 2325K and memory cost of 80KB which seems reasonable at the spec of 1080P@90fps.

5.2 Future Work

In the future, we will focus on both disparity map quality enhancement and also costing down the hardware cost to make it more economy-efficient. We can add motion vectors as input factors on both disparity propagation stage and temporal consistency stage to make sure the disparity surface is smoother and the flickering problem is less. We can also use downsample technique to reduce the gate count and memory cost of hardware to make it more hardware-friendly.

Reference

- [1] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling" *IEEE Trans. Pattern Anal. Mach. Intell.(TPAMI)*, no. 20, vol. 4, pp. 401-406, Apr. 1998.
- [2] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. on Comput. Vision Pattern Recognition (CVPR'07)*, Jun. 2007.
- [3] N. Y.-C. Chang, Y.-C. Tseng, and T.-S. Chang, "Analysis of color space and similarity measure impact on stereo block matching," in *Proc. IEEE Asia Pacific Conf. on Circuits and Syst. (APCCAS'08)*, Dec. 2008, pp. 926-929.
- [4] J. Lu, G. Lafuit, and F. Catthoor, "Anisotropic local high-confidence voting for accurate stereo correspondence," in *Proc. SPIE Image Process.: Algorithm and Syst. VI*, vol. 68120, Jan. 2008.
- [5] K. Zhang, J. Lu, and G. Lafuit, "Scalable stereo matching with locally adaptive polygon approximation," in *Proc. IEEE Int. Conf. on Image Process. (ICIP'08)*, Oct. 2008, pp. 313-316.
- [6] K. Zhang, J. Lu, and G. Lafuit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, no. 19, vol. 7, pp. 1073-1079, Jul. 2009.
- [7] K.-J. Yoon and I.-S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650-656, Apr. 2006.
- [8] M.-H. Ju and H.-B. Kang, "Constant time stereo matching" in *Proc. Int. Conf. on Machine Vision and Image Process. (IMVIP'09)*, Sep. 2009, pp. 13-17.

- [9] W. Yu, T. Chen, F. Franchetti, and J. C. Hoe, "High performance stereo vision designed for massively data parallel platforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1509-1519, Nov. 2010.
- [10] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 792-805, Jun. 2010.
- [11] Y. Ohta and T. Kanade, "Stereo by intra- and inter- scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, no. 7, vol. 2, pp. 139-154, Mar. 1985.
- [12] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *Proc. IEEE Conf. on Comput. Vision Pattern Recognition (CVPR'05)*, 2005, pp. 384-390.
- [13] Y. Deng and X. Lin, "A fast line segment based dense stereo algorithm using tree dynamic programming," in *Proc. European Conf. on Comput. Vision (ECCV'06)*, 2006, pp. 201-210.
- [14] C. Lei, J. Selzer, Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *Proc. IEEE Conf. on Comput. Vision Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2378-2385.
- [15] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. IEEE Int. Conf. on Comput. Vision (ICCV'01)*, vol. 2, Jul. 2001, pp. 508-515.
- [16] L. Ford and D. Fulkerson, *Flows in networks*, Princeton Univ. Press, 1962.
- [17] A. V. Goldberg, "A new approach to the maximum flow problem," *J. of the ACM*, vol. 35, pp. 921-940, 1988.

- [18] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.
- [19] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 26, no. 9, pp. 1124-1137, Sep. 2004.
- [20] C.-W. Chou, J.-J. Tsai, H.-M. Hang, and H.-C. Lin, "A fast graph cut algorithm for disparity estimation," in *Proc. Picture Coding Symp. (PCS'10)*, Nagoya, Japan, Dec. 2010, pp. 326-329.
- [21] B. V. Cherkassky and A. V. Goldberg, "On implementing the push-relabel method for the maximum flow problem," *Algorithmica*, New York Inc.: Springer-Verlag, 1997, vol. 19, pp. 390-410.
- [22] A. Delong and Y. Boykov, "A scalable graph-cut algorithm for N-D grids," in *Proc. IEEE Conf. on Comput. Vision Pattern Recognition (CVPR'08)*, Jun. 2008.
- [23] N. Y.-C. Chang and T.-S. Chang, "A scalable graph-cut engine architecture for real-time vision," in *Proc. VLSI design/CAD Symp.*, Hualien, Taiwan, 2007.
- [24] J. Sun, N.-N. Zhang, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 25, no. 7, pp. 787-800, Jul. 2003.
- [25] P. F. Felzenswalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vision (IJCV)*, vol. 70, no. 1, pp. 41-54, May 2006.
- [26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov

Random Fields with smoothness-based priors,” *IEEE Trans. Pattern Anal. Mach. Intell.* (TPAMI), vol. 30, no. 6, pp. 1060-1080, Jun. 2008.

[27] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, “Real-time global stereo matching using hierarchical belief propagation,” in *Proc. British Mach. Vision Conf. (BMVC)*, 2006.

[28] S. Park C. Chen, and H. Jeong, “VLSI architecture for MRF based stereo matching,” in *Proc. Int. Symp. on Syst., Architecture, Modeling and Simulation (SAMOS’07)*, Greece, Jul. 2007.

[29] C.-C. Cheng, C.-K. Liang, Y.-C. Lai, H. H. Chen, and L.-G. Chen, “Analysis of belief propagation for hardware realization,” in *Proc. IEEE Workshop on Signal Process. Syst. (SiPS’08)*, Washington DC, USA, Oct. 2008, pp. 152-157.

[30] C.-C. Cheng, C.-K. Liang, Y.-C. Lai, H. H. Chen, and L.-G. Chen, “Fast belief propagation process element for high-quality stereo estimation,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP’09)*, Taipei, Taiwan, Apr. 2009, pp. 745-748.

[31] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H. H. Chen, “Hardware-efficient belief propagation,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR’09)*, Florida, USA, Jun. 2009, pp. 80-87.

[32] C.-C. Cheng, C.-T. Li, C.-K. Liang, Y.-C. Lai, and L.-G. Chen, “Architecture design of stereo matching using belief propagation,” in *Proc. IEEE Int. Symp. Circuits and Syst. (ISCAS’10)*, Jun. 2010, pp. 4109-4112.

- [33] C.-K. Liang, C.-C. Cheng, Y.-C. Li, L.-C. Chen, and H. H. Chen, "Hardware-efficient belief propagation," *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, vol. 21, no. 5, pp. 525-537, May 2011.
- [34] S. C. Park and H. Jeong, "Memory-efficient iterative process for two-dimensional first-order regular graph," *Optics Letter*, vol. 33, no. 1, pp. 74-76, Jan. 2008.
- [35] T. Yu, R.-S. Lin, B. Super, B. Tang, "Efficient message representation for belief propagation," in *Proc. IEEE Int. Conf. on Comput. Vision (ICCV'07)*, Oct. 2007.
- [36] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and self-adapting dissimilarity measure," in *Proc. IEEE Int. Conf. on Pattern Recognition (ICPR'06)*, Sep. 2006, pp. 15-18.
- [37] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 31, no. 3, pp. 1-13, Mar. 2009.
- [38] E. S. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs, "Temporally consistent reconstruction from multiple video streams using enhanced belief propagation," in *Proc. IEEE Int. Conf. on Comput. Vision (ICCV'07)*, Rio de Janeiro, Brazil, Oct. 2007.
- [39] G. Egnal and R. P Wildes, "Detecting binocular half-occlusions: empirical comparisons of five approaches," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 24, no. 8, pp. 1127-1133, Aug. 2002.
- [40] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to analysis and automated cartography," *Commun. of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

- [41] M. Gong, "Enforcing temporal consistency in real-time stereo estimation," in *Proc. European Conf. on Comput. Vision (ECCV'06)*, vol. 3953, 2006, pp. 564-577.
- [42] D. Min, S. Yea, Z. Arican, and A. Vetro, "Disparity search range estimation: forcing temporal consistency," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP'10)*, Dallas, Texas, May 2010, pp. 2366-2369.
- [43] R. Khoshabeh, S. H. Chan, T. Q. Nyuyen, "Spatio-temporal consistency in video disparity estimation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP'11)*, Prague, Czech Republic, May 2011.
- [44] *Depth estimation reference software (DESR)*, version 4.0 [Online]. Available: http://wg11.sc29.org/svn/repos/MPEG-4/test/tags/3D/depth_estimation/DESR_4
- [45] Enhancement of temporal consistency for multi-view depth map estimation, ISO/IEC JTC1/SC29/WG11, M15594, Jul. 2008.
- [46] Depth estimation improvement for depth discontinuity areas and temporal consistency preserving, ISO/IEC JTC1/SC29/WG11, M16048, Feb. 2008.
- [47] The consideration of the improved depth estimation algorithm: the depth estimation algorithm for temporal consistency enhancement in non-moving background, ISO/IEC JTC1/SC29/WG11, m16070, Jan. 2009.
- [48] D. Scharstien and R. Szeliski, *Middlebury Stereo Evaluation – Version 2* [Online]. Available: <http://vision.middlebury.edu/stereo/eval/>
- [49] Q. Wei, "Converting 2D to 3D: a survey," Inform. and Commun. Theory Group, Faculty Elect. Eng., Math. and Comput. Sci., Delft Univ. of Technol., Netherlands, Research Assignment, Dec. 2005.

- [50] D. Hoiem, A. Stein, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from a single image," in *Proc. IEEE Int. Conf. on Comput. Vision (ICCV'07)*, Oct. 2007.
- [51] D. Hoiem, A. Efros, and M. Hebert, "Recovering surface layout from an image," *Int. J. Comput. Vision (IJCV)*, vol. 75, no. 1, pp. 151-172, Oct. 2007.
- [52] D. Scharstien and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithm," *Int. J. Comput. Vision (IJCV)*, vol. 47, no. 1-3, pp. 7-42, May 2002.
- [53] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 25, no. 8, pp. 993-1008, Aug. 2003.
- [54] Lu. Zhang, "Design and Implementation of Real-Time High-Definition Stereo Matching SoC on FPGA"
- [55] Yu-Cheng Tseng, "The Study of Disparity Estimation Design for High Definition 3DTV Application"

