國立交通大學

資訊科學與工程研究所

碩士論文

基於區域可變式視窗大小和
適應性權重的視差估計演算法

Region-Based Variable Window Size with Adaptive Support Weight for Disparity Estimation Algorithm

研 究 生：黃致遠

指導教授：蔡文錦　教授

中華民國 101 年 12 月

基於區域可變式視窗大小和適應性權重的視差估計演算法

Region-Based Variable Window Size with Adaptive Support Weight for Disparity Estimation Algorithm

研究生：黃致遠　　Student：Chih-Yuan Huang

指導教授：蔡文錦　　Advisor：Wen-Jiin Tsai

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

December 2012

Hsinchu, Taiwan, Republic of China

中華民國 101 年 12 月

中文摘要

立體視差估計演算法被廣泛的利用在許多實際應用層面，像是3D視訊會議和多視角立體電視等。在一個典型的地區式視差估計演算法當中，通常會使用一個固定的視窗大小來聚合視差匹配的代價。然而，越大的視窗大小雖然能提升在低紋理區塊的表現，卻也同時讓視差的邊界被模糊化。

在本篇論文中，我們提出了一個可變動的視窗大小，先利用顏色資訊把圖片連結或切割成多個區塊，再利用區塊的資訊來決定視窗的大小，同時在最後的優化步驟，這些資訊也能用在一個十字區域式表決機制。另外，為了讓計算結果的品質更加提升，我們更結合了微型普查(mini-census)和顏色差距的比對方式。從實驗結果顯示，我們的方法能夠有效提升原本方法的效能，而且經由Middlebury網站的評估，我們的方法是目前的地區式演算法當中名列前茅的。

關鍵字：視差估計、可變動視窗大小、區塊分割

ABSTRACT

Stereo matching algorithm has been widely adopted by various stereo vision applications such as 3D video conference and free viewpoint TV. In the typical local methods for stereo matching, a fixed support window size is often adopted in cost aggregation step. Larger supporting window can improve the stereo matching performance at low texture regions. However, it is blurred near depth discontinuities.

In this paper, we propose a variable window size selection strategy before cost aggregation step. The strategy determines the support window by utilizing the segment information derived from a color based segmentation method. This information is also used for region-based cross voting scheme in refinement step. Moreover, a combined matching cost measure with mini-census and color difference is proposed. Experimental results show that the proposed method effectively improves the performance of original method. According the performance evaluation at the Middlebury website, the proposed method is one of the current state-of-the-art local methods.

Index Terms-- Stereo matching 、Variable support window 、image segmentation

誌謝

在這兩年多的研究所生涯中，能完成我的碩士論文，首先最要感謝的就是我的指導教授蔡文錦博士。在學業研究上，孜孜不倦地與我討論各種相關的議題，點出我研究上的盲點，引導我前往正確的方向；在日常生活中，不時的關心我並且給予我前進的力量。在此向我最敬愛的指導教授蔡文錦博士，致上最高的敬意。

我要感謝實驗室的學長姐，吳佳穎、呂威漢、張育誠、謝寧靜、孫域晨、詹家欣，謝謝你們指導我各種研究上的相關知識。另外要感謝我的同學們，蕭成憲、楊巧安、林宗翰，謝謝你們陪伴我度過這段追求知識的過程，課程上的互相砥礪，生活中的互相打氣，讓我從你們身上獲益良多。還有謝謝學弟們，王敬嚴、林建儒、胡振達、高彬倬、邱柏瑞、劉翊士，謝謝你們讓我在研究所生涯中過得更加精彩，祝福你們順利畢業。

最重要的，感謝我的家人，尤其是我的父母親，在背後默默支持我，讓我在疲憊的時候有一個溫暖的避風港，謝謝你們對我的期待及付出。

接下來就要告別學生生涯進入職場了，大家珍重。

謹以此論文獻給我的師長、家人及所有關心我的朋友們

CONTENTS

LIST OF FIGURE

LIST OF TABLE

Chapter 1 Introduction

Stereo matching is one of the most actively studied topics in computer vision. The issue is to estimate the disparity map from a pair of rectified images of the same scene taken from different viewpoints. The disparity of a pixel is the displacement vector between corresponding pixels which horizontally shift from the left image to the right image. The process of finding the disparity is referred as *stereo matching* or *disparity estimation*. In recent years, a large number of algorithms have been proposed to solve the problem. The stereo matching algorithm has been widely adopted by applications such as 3D video conference and free viewpoint TV [1]. It will continue to be an attractive topic with the development of 3D video market in the next few years.

According to the work published by Scharstein and Szeliski [2], a stereo matching algorithm generally consists of the following four steps: matching cost computation, cost (support) aggregation, disparity computation and disparity refinement. **Figure 1-1** depicts the flow of a stereo matching algorithm.

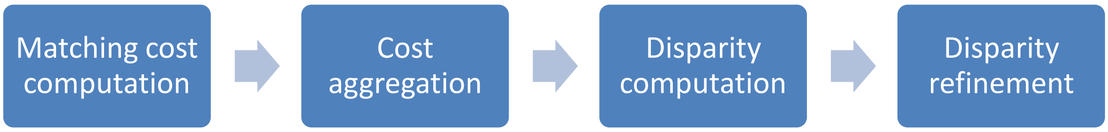


Figure 1-1 A general flow of stereo matching algorithm.

The first step computes the initial matching costs of all disparity candidates for each pixel by the cost measure, such as absolute difference (AD), gradient-based measures, and non-parametric transforms like rank and census [3]. Among these measures, the AD cost is the most commonly used for many stereo matching methods due to its simplicity. The AD cost in RGB color space of a pixel p with respect to a disparity d can be defined as:

$$C_{AD}(p,d) = \frac{1}{3}\sum_{i\in\{R,G,B\}}\left|I_L{}^{i}(p) - I_R{}^{i}(q)\right| \tag{1}$$

Where $\mathrm{I_L}$ and $\mathrm{I_R}$ represent the left and right images. The AD cost evaluating the matching penalty seems intuitive; however, it has poor quality for global radiometric changes. In a recent experiment by Hirschmüller and Scharstein [4], the census transform performs the best overall results in stereo matching methods, since the match metrics compare the relative orderings instead of the intensity of the pixels.

Second, the cost aggregation step gathers the costs in a support window which is usually a square window. A simple hypothesis of the cost aggregation is that surrounding pixels with similar colors should be greatly correlated to the center pixel. An aggregated cost of pixel p can be calculated by the following function:

$$C_{agg}(p,d) = \frac{\sum_{i\in W_L} cost(i,d) \times w(p,i)}{\sum_{i\in W_L} w(p,i)} \tag{2}$$

Where $\mathrm{cost(i)}$ means the initial matching cost obtained from the previous step. In addition, $\mathrm{w(p,i)}$ represents the related weight between the pixel p and its neighbors i in current support windowW_L. A simple schematic diagram of cost aggregation is depicted in **Figure 1-2**. The cost aggregation reduces the matching ambiguities and noise in the initial cost volume due to lack of more information.

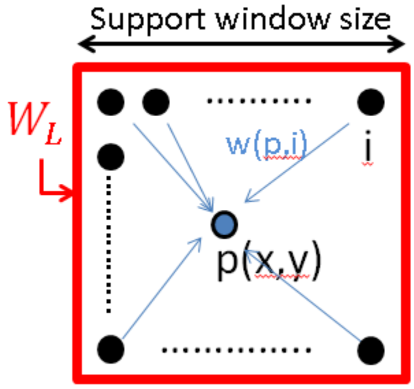


Figure 1-2 A cost aggregation in pixel p with its neighbors i

With the aggregated costs, the disparity map can be simply computed by the winner-take-all (WTA) process, which is to select the disparity candidate with the minimal aggregated cost. The WTA method can be expressed as

$$D(p) = \arg\min_{d} C_{agg}(p, d) \tag{3}$$

, where d is the disparity candidate over a disparity search range. Another optimization method such as graph-cut is introduced by Boykov et al. in [5].

Finally, the disparity refinement step further refines the disparity by correcting the error caused by the outlier pixels or image noise. A simple refinement tool is a 3x3 median filter. **Figure 1-3** represents how a median filter works. In addition, left-right consistency check [6] is an effective refinement technique to deal with the occluded region. We will introduce it later in 3.7.1.

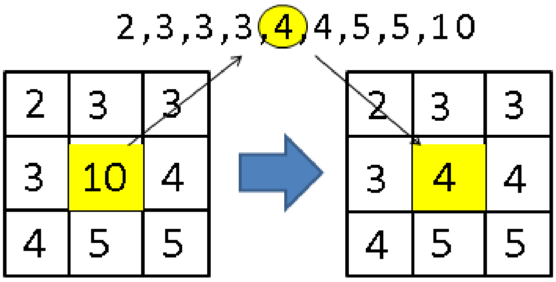


Figure 1-3 A 3x3 median filter

With the above steps, the disparity estimation algorithms can be roughly classified into two types: local algorithms and global algorithms. Local methods focus on the matching cost computation step and the cost aggregation step. Instead, global methods emphasize on the disparity computation step.

Local algorithms estimate the disparity of each pixel independently within a support window. The matching costs are aggregated over the window, after that, the disparity candidate with the minimal cost is simply selected for the pixel by the WTA rule. The local algorithms have low computation complexity and storage requirement, so they are generally adopted by real-time applications. Recent research has proved that a well-selected support window can give a quality result. The local approach with the adaptive support weight (ADSW) is proposed by Yoon et al. [7] , which can achieve the goal to apply a support window of arbitrary shape. Therefore, the ADSW can have the comparable result approaching to the global method with considerable execution time. Later, Tombari et al. proposed a segment-based support method [8] to assign different weights to the pixels in a support window by based on the segmentation information. Recently, the mini-census adaptive support weight approach (MCADSW) [9] is accomplished to have lower complexity and more capability of handling brightness bias problems than the original ADSW.

On the other hand, the cost aggregation step is simple in global algorithms. Instead, the emphasis is on the disparity computation step. Global approaches formulate the stereo matching problem as the objective energy function and minimize it to determine the disparity map. The energy functions often include data term and a neighboring term. Some efficient optimizers like graph-cut [5] and belief propagation are employed to minimize the energy function. A cooperative optimization based on region segment proposed by Wang and Zheng [10] is the state-of-art of global approach according to the Middlebury evaluation [11]. This algorithm uses regions as matching objects and defines the corresponding energy function with the constraints on data term, smoothness, and occlusion. Consequently, global methods produce more accurate results than common local methods, but they suffer from high computational complexity and can hardly be used in real-time implementation.

The proposed algorithm in this thesis is focused on local algorithms. We modified the MCADSW [9] algorithm by adopting a robust matching cost measure and utilizing the segment information to determine variable support size for cost aggregation. Moreover, the region information can also be used in our region-based cross voting scheme to improve the quality of the disparity refinement result.

The rest of this thesis is organized as follows. Section 2 introduced the details of the related works about local stereo matching methods. Section 3 describes our motivation and the proposed algorithm step by step. The experimental results are presented in section 4. Finally, section 5 concludes this thesis.

Chapter 2 Related Work

2.1 Adaptive Support Weight Approach

Yoon proposed the concept of adaptive support weight (ADSW) [7] that assigned different weights to the neighboring pixels in a support window. Two main grouping concepts in support-weights generation are proximity and similarity. The former is related to the spatial distance between surrounding pixels, and the latter is related to the color distance of two pixels. Based on above principles, the weight function in the cost aggregation equation (2) can be defined as

$$\mathrm{w(p,i)} = \exp\left(-\left(\frac{\mathrm{d_s(p,i)}}{\lambda_\mathrm{s}} + \frac{\mathrm{d_c(p,i)}}{\lambda_\mathrm{c}}\right)\right) \tag{4}$$

,where $\mathrm{d_s}$ and $\mathrm{d_c}$ are the Euclidean distance respectively the distance between two coordinates and the distance in CLELAB color space. λ_s and λ_s are two constants. The cost aggregation strategy in ADSW is symmetrical, and the support weights in both left and right image are generated and taken into account.

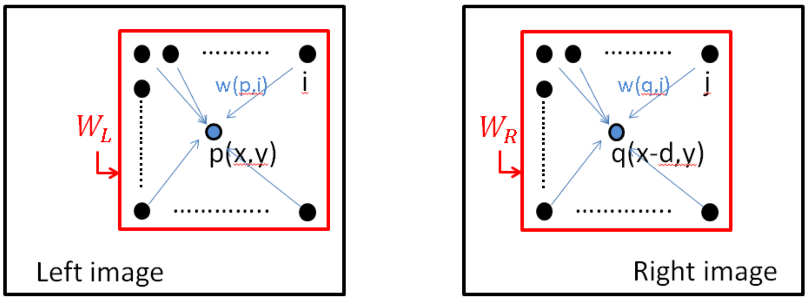


Figure 2-1 A symmetrical cost aggregation

Figure 2-1 describes the symmetrical cost aggregation diagram. Let p and q be respectively the center pixels in the current support window W_L in the left image and the support window W_R in the right image. Then, the cost aggregation is rewritten as following

$$C_{agg}(p,d) = \frac{\sum_{i\in W_L, j\in W_R} TAD(i,j)\times w(p,i)\times w(\mathrm{q,j})}{\sum_{i\in W_L, j\in W_R} w(p,i)\times w(q,j)} \qquad (5)$$

For any point $\mathrm{i} \in W_L$corresponding to $\mathrm{j} \in W_R$, the matching cost is computed by using the Truncated Absolute Difference (TAD), which can be expressed as

$$TAD(i,j) = \min\left\{\sum_{k\in\{R,G,B\}} \left|{I_L}^k(i) - {I_R}^k(j)\right|, T\right\} \qquad (6)$$

,where T is the truncation threshold parameter to reject outliers. After the dissimilarity computation, the disparity map is computed by Winner-Takes-All (WTA) strategy. The adaptive support weight gives an excellent performance in low-textured area and near depth discontinuities. And the disparities in occluded regions can be corrected by the left-right consistency check.

2.2 Segmentation-Based Adaptive Support Weight

In [8], Tombari et al. analysis the result obtained by the ADSW method. There are some drawbacks in high-textured surfaces and repetitive patterns, since the amount of aggregated support in ADSW may be insufficient in these regions. This is because the use of spatial distance would determine the wrong support.

The basic idea of segmentation-based adaptive support is to embody the concept of color similarity as well as segmentation information to improve the capability of the support window. The proximity term has been eliminated from the support weight computation. Instead, it is assumed that the disparity of each pixel has a similar value in the same segment which can be obtained from a segmentation process, such as Mean-Shift algorithm [12]. Hence, a novel weight function is proposed as follows:

$$w(p,i) = \begin{cases} 1.0 & i \in Segment(p) \\ \exp(-\frac{d_c(p,i)}{\lambda_c}) & otherwise \end{cases} \tag{7}$$

The weight has to be the maximum value if the pixel i lying on the same segment of the central pixel p, otherwise, it is computed based on color similarity. The Segment-support method provides notable improvements rather than only relying blindly on spatial proximity and color similarity.

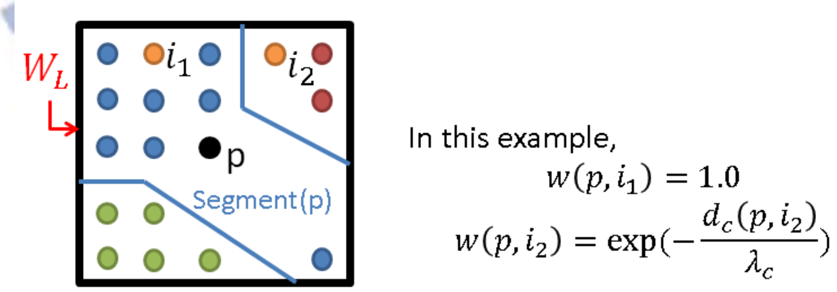


Figure 2-2 A example for the segmentation-based adaptive weight generation

2.3 Mini-Census Adaptive Support Weight

The mini-census adaptive support weight method (MCADSW) [9] is an effective algorithm modified from the ADSW method, and the former one has much lower complexity and more capability of dealing with brightness problem. The MCADSW adopted the mini-census transform to improve the robustness to radiometric distortion. Because mini-census cost measure has only relative information, a scale-and-truncated approximation of the weight function is proposed. In addition, the two-pass approach not only reduces computation complexity but also derives good performance in low-texture areas.

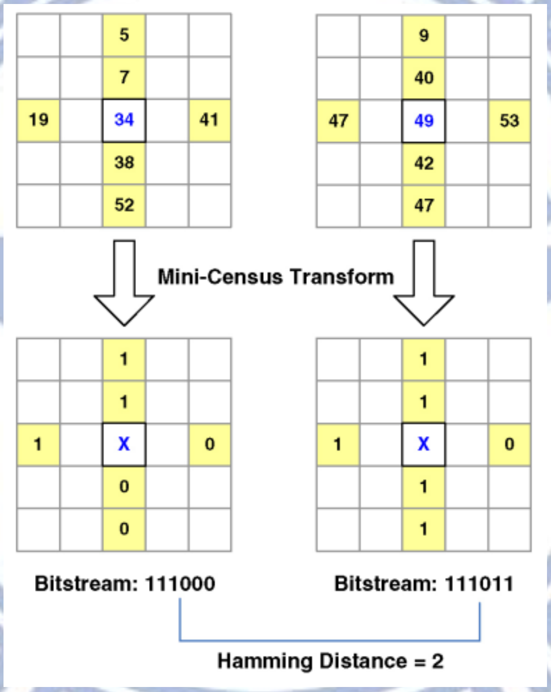


Figure 2-3 The mini-census transform and matching introduced in [9]

2.3.1 Mini-Census Transform and Matching

The concept of original census transform [3] is to obtain relative information instead of the intensity itself. The intensity used to compare is first transformed into Y color space. If a pixel’s intensity is larger than the center pixel’s intensity, it is transformed into the label 0, otherwise the label 1. The mini-census transform compares the intensity of the 6 significant pixels with the center pixel. After the transformation, each pixel can be represented as a 6-bit binary bitstream. The mini-census cost between the corresponding pixels is taken as the hamming distance between two mini-census bitstreams, which is defines as

$$C_{mc}(p,d) = Ham\big(b_L(p), b_R(p-d)\big) \qquad (8)$$

,where C_{mc} represents the mini-census cost of pixel p with the disparity level d. Ham() is the hamming distance function. $b_L(p)$ and $b_R(p-d)$ are referred as the bitstream of current pixel in the left image and the bistream of corresponding pixel in the right image, respectively. An example of the mini-census transform and matching introduced in [9] is shown in **Figure 2-3**. The mini-census cost performs better disparity result in occluded area than the traditional SAD does.

2.3.2 Weight Generation

The weight function in the MCADSW method also eliminates the proximity weight. Moreover, a scale-and-quantize weight function is defined as

$$w(p,i) = quantize\left[exp\left(-\frac{d_c(p,i)}{\lambda_c}\right) \times scaling\ factor\right] \qquad (9)$$

The weight function is scaled up by 64 and quantized to leave only one nonzero most significant bit. The purpose of scaling the function is to increase the influence of neighbor pixel which has the small color distance.

2.3.3 Two-pass Cost Aggregation

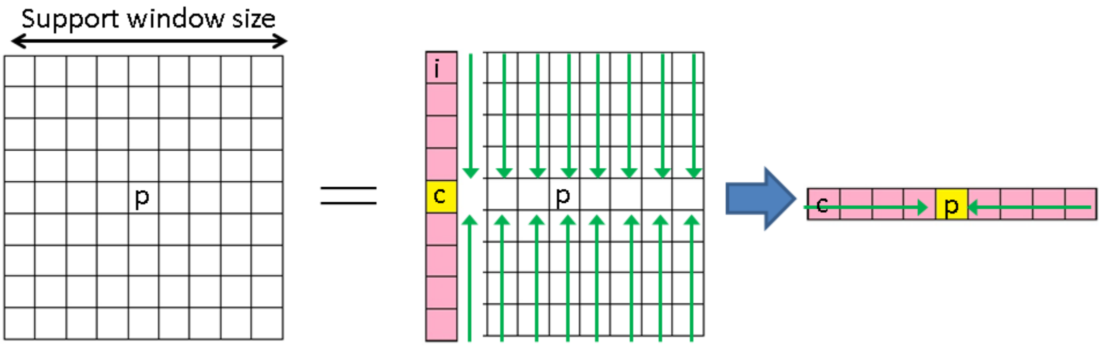
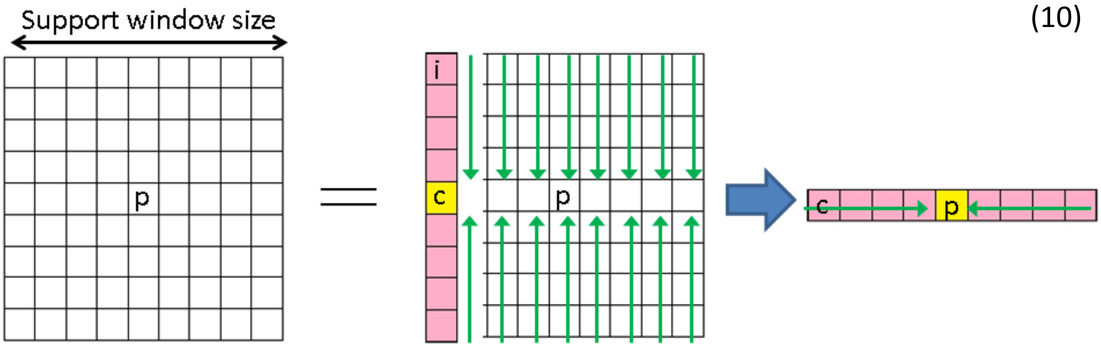


Figure 2-4 Two-pass cost aggregation

The original cost aggregation (5) is divided into vertical and horizontal cost aggregation in a two-pass approach. First, the matching cost aggregates vertically to compute a vertical cost of each column, and then aggregate each cost of the column in the support window horizontally to compute the final aggregated cost. The equations are written as equations (10) and (11), and are depicted in **Figure 2-4**.

(10)



$$Cost_{col} = \sum_{i \in col} C_{mc}(i,d) \times w(i,c)$$

$$C_{agg}(p,d) = \sum\nolimits_{col \in W_L} Cost_{col} \times w(p,c) \tag{11}$$

where C_{mc} is the mini-census cost defined in equation (8). Notice that c represents the center pixel which changes column by column. Consequently, the MCADSW method may improve the low-textured region with smoothing disparity, since the correlated weight of the border or corner pixels could be increased with the help of closer center pixel.

Chapter 3 Proposed Method

3.1 Motivation

In [7], the adaptive support weight can achieve the effect of using support window with arbitrary sharp and give a quality depth result. Later, [2] proposed a segment-based support weight to improve the performance. Mini-census transform in [3] is employed to improve the capability of handling the lighting effect. However, they all used a fixed support window size. According to our observation in Figure 3-1, , it shows that the performance will not always be better while the support window size gets larger. This drawback is evident especially for *Teddy* and *Cones* sequences. The issue of support window size still remains. Moreover, the mini-census matching cost in [3] is much less sensitive to brightness bias, but it might cause errors in low textured region since it lacks of color information.

According to the problems mentioned above, we proposed an algorithm which is modified from the Mini-Census Support Weight (MCADSW) [3]. In the proposed method, we combine the mini-census transform with the robust absolute differences (RAD) measure to increase the robustness of the matching process. In addition, the segment-based idea is inspired from [2] which apply the information obtained from the segmentation within the cost aggregation step. With the additional information, we could determine the variable support window size instead of a fixed one. Besides, the segment information can also be used for disparity refinement to improve the performance of the final disparity result.

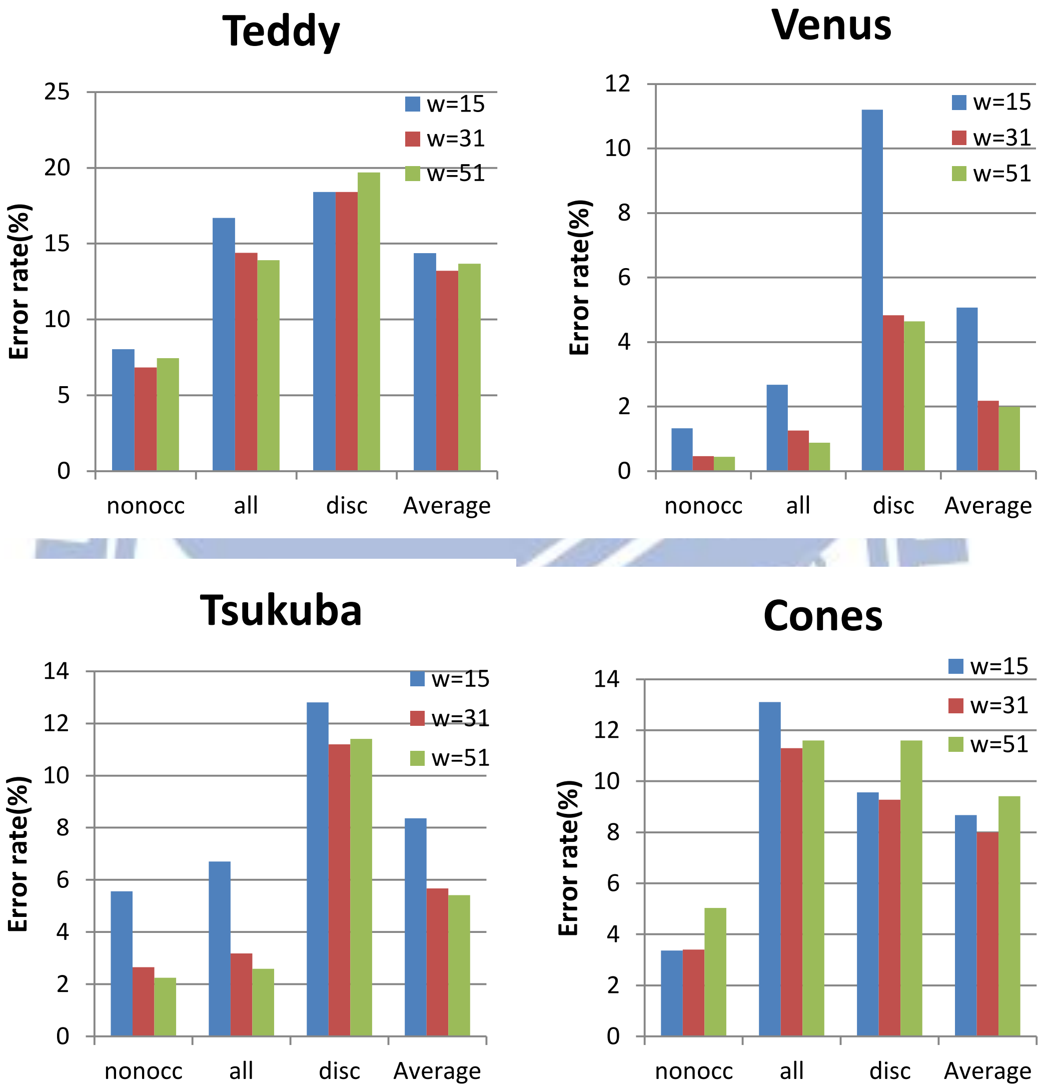


Figure 3-1 The performance comparison with different window size

3.2 The flow of the proposed method

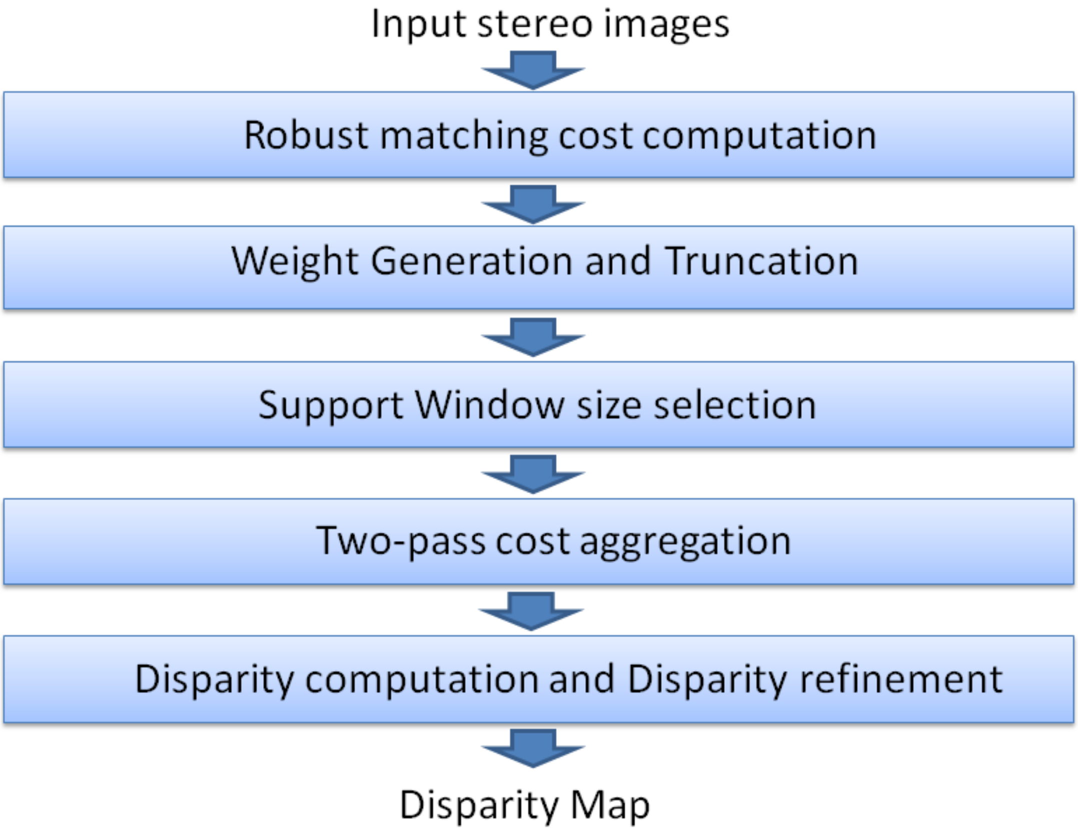

Input stereo images
Robust matching cost computation
Weight Generation and Truncation
Support Window size selection
Two-pass cost aggregation
Disparity computation and Disparity refinement
Disparity Map

Figure 3-2 The flow of the proposed algorithm

Figure 3-2 shows the flow of the proposed algorithm. First, we employ the Mean-Shift algorithm to segment the inputted stereo images into regions according to similar color. Second, the initial matching cost is computed by the combined matching cost measure. After that, the weight generation and truncation step generates the weight coefficients needed in the cost aggregation. Before the cost aggregation, we employ the segment information to determine the support window size. And then, the matching cost will be aggregated by a two-pass cost aggregation. Once the aggregated cost is computed, the disparity map can be simply computed through a winner-take-all

(WTA) strategy. Finally, the result can be refined by left-right check and region-based cross voting.

3.3 Robust matching cost computation

Mini-census cost transforms the intensity of the data term into relative bitstreams. Hence it can tolerate outliers caused by radiometric noise. Nevertheless, it could also lead to matching ambiguities in the regions with repetitive pattern, while the color information can deal with these matching ambiguities. The idea of combining cost measures for improved performance of matching process is inspired from the works accomplished by Mei et al [13]. . They propose a combined cost measure with the AD and census for matching cost initialization. The benefit of the combination is impressive with a few additional computation time.

Based on MCADSW [9], we tend to preserve the mini-census cost measure $\mathrm{C_{mc}}$ and combine with the color constraint. However, the origin AD cost range [0,255] is much bigger than the mini-census cost range [0,6], so the combination cost might be biased to the AD cost. Therefore, a robust function R is adopted for the AD measure which maps the cost value range from [0,255] to [0,1]. It is defined as

$$\mathrm{R}(\mathrm{C_{AD}}, \lambda_{\mathrm{AD}}) = 1 - \exp(-\frac{\mathrm{C_{AD}}}{\lambda_{\mathrm{AD}}}) \tag{12}$$

,where $\mathrm{C_{AD}}$ is the traditional AD cost described as equation (1), and λ_{AD} is allowed to control the influence of outliers. Given a pixel p with respect to a disparity level d, the proposed robust matching cost can be calculated as follows:

$$C_{Robust}(p,d) = C_{mc} + \lambda_m \times R(C_{AD}, \lambda_{AD}) \tag{13}$$

,where λ_m is a tuning constant to control the influence between color similarity and the relative information.

3.4 Weight generation and truncation

We exploit the weight function of the MCADSW described in equation (9), In addition, a simple truncation is implemented to alleviate the matching error due to outlier pixels. We give a minimum value zero to these outliers. Hence, a modified weight function is proposed:

$$w(p,i) = \begin{cases} 0 & , if \;\; d_c(p,i) > T_w \\ quantize\left[exp\left(-\frac{d_c(p,i)}{\lambda_c}\right) \times scaling\ factor\right] & , otherwise \end{cases} \tag{14}$$

,where T_w represents the threshold to separate the outliers; the cost $d_c(p,i)$ is sum of color difference in YUV color space which is defined as

$$d_c(p,i) = \sum_{k \in \{Y,U,V\}} \left| I_k\,(p) - I_k\,(i) \right| \tag{15}$$

The curve of our weight function is shown in **Figure 3-3**.

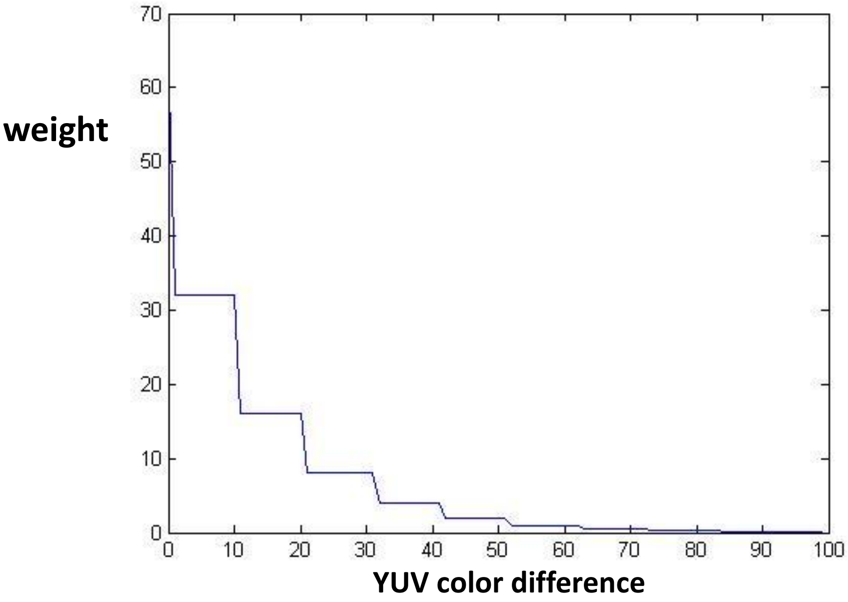


Figure 3-3 The curve of our proposed weight function

3.5 Support window size selection

The basic concept of the proposed method is to employ the segment information to change the correlation window size. From our analysis, the MCADSW method still suffers from some incorrect disparity estimation at occlusion, low textured, and repeating pattern areas. Although we can improve the matching quality by applying a larger supporting window, the disparity map might be blurred near the discontinuous area.

3.5.1 Image segmentaion

The concept of the segmentation process is to cut the image into several regions which can enlarge the region size as big as the spatial and color is assessed. In our approach, we adopt the Mean-Shift algorithm to process the segmentation. Two constant parameters σ_S and σ_R are referred as *spatial radius* and *range radius* which construct the restraint of growing the segment. And the parameter $Fuse_R$ promises the minimum size of each region. **Figure 3-4** shows the left image and its segmentation result of each of four testing stereo images available on Middlebury website [11], with the same parameter set: $\sigma_S = 3,\ \sigma_R = 3,\ Fuse_R = 35.$

After the image segmentation step, the addition information of segment represents an intelligent proximity rather than the spatial distance, and it can be employed later in the support window size selection and the refinement step.

3.5.2 Variable window size selection

Instead of adopting a fixed support window size, the proposed method makes use of information obtained from the Mean-shift algorithm to determine the support window size. By observing the segment result shown in **Figure 3-4**, a large segment is often related to a low textured or repeating area, where the support window should be enlarged to obtain more support information. On the other hand, a small segment is usually related to a border or occluded area. The MCADSW method can achieve the window with arbitrary shape to deal with border area, and the occluded area can be refined later by the refinement step.

Hence, the window size selection can be expressed as:

$$W = \begin{cases} W_{small} & , if\ Num(Seg(p)) < T_{count} \\ W_{big} & , otherwise \end{cases} \tag{16}$$

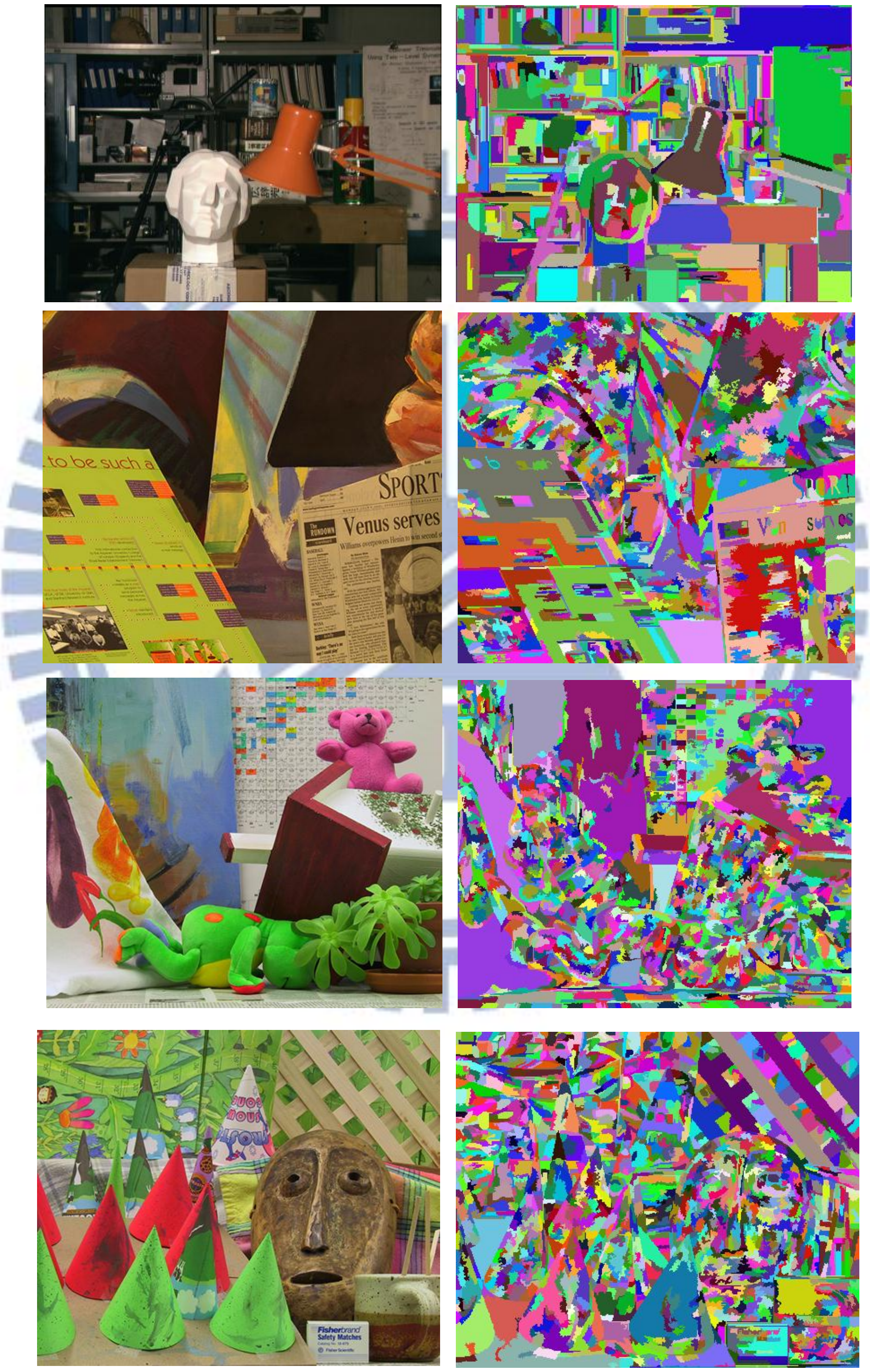


Figure 3-4 The left image and the segmentation result of 4 stereo images

3.6 Two-pass cost aggregation

Once the window size is selected, the two-pass cost aggregation can be processed. We adopt the same aggregation direction in [9] but simply modify the original aggregation function (10)(11) with normalization after the cost aggregates. The modified functions are described as equations (17) and (18):

$$Cost_{col} = \frac{\sum_{i \in col} C_{Robust}(i,d) \times w(i,c)}{\sum_{i \in col} w(i,c)} \tag{17}$$

$$C_{agg}(p,d) = \frac{\sum_{col \in W} Cost_{col} \times w(p,c)}{\sum_{col \in W} w(p,c)} \tag{18}$$

3.7 Disparity refinement

The disparity maps computed by the above steps contain errors in the occluded and discontinuous regions. The results of left and right images are denoted as D_L and D_R, respectively. We refine the disparity errors by several steps. First, we smoothen the disparity map by a 3x3 median filter to alleviate the noise. Second, we detect the outliers by the left-right check and then classify them into two categories. After that, we treat the two kinds of outliers with different refinement rules. Finally, a region-based cross voting refinement is performed. The first step has been introduced in **Figure 1-3**, and the other steps are introduced as following.

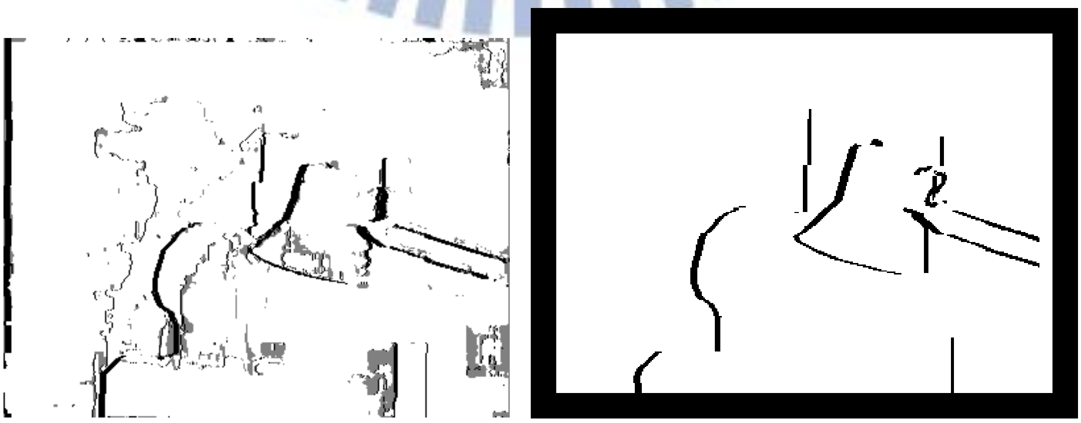
3.7.1 Left-right check and Outliers classification

Left-right consistency check [6] is a widely used technique to detect the outliers. For each pixel, a following check is performed:

$$D_L(p) = D_R(p - (D_L(p), 0)) \tag{19}$$

If a pixel can’t satisfy the check, it is considered as an outlier. And then, based on the method proposed by Hirschmϋller [14], we classify the outliers into “occlusion” and “mismatch” points. For each outlier pixel p, we perform the left-right check with all disparity candidates, and if no candidate could hold the check, p is an “occlusion” pixel, otherwise a “mismatch” pixel.

Figure 3-5 demonstrates the classification results of four sequence on Middlebury data sets compared with the occluded-region in the ground truth. If a pixel is labeled as “occlusion”, we depict it with black color, and a “mismatch” pixel is depicted with gray. The classification can effectively separate the occluded-region which is similar to the ground truth.



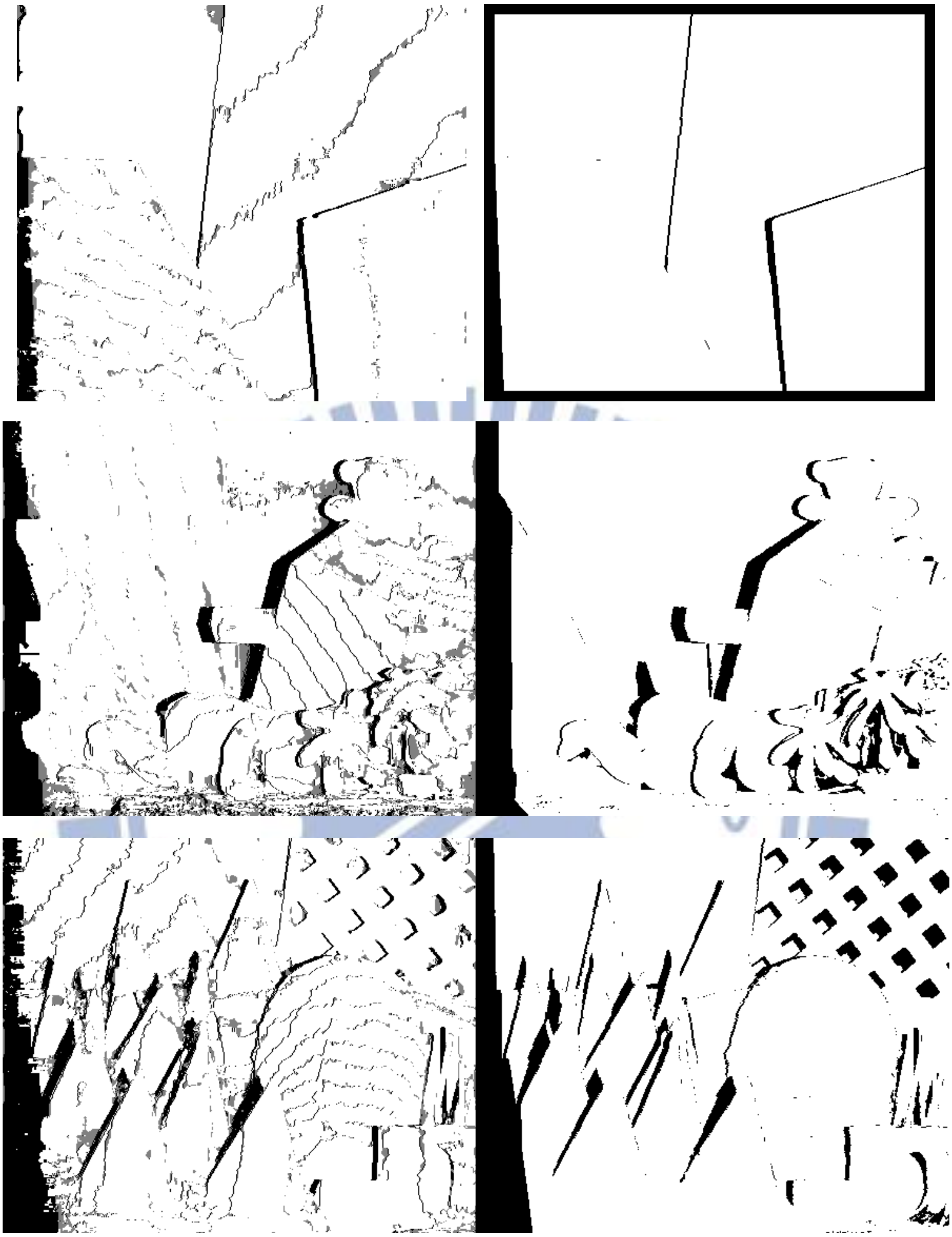


Figure 3-5 The classification results(left) and the occluded-region in the ground truth(right)

3.7.2 Outlier refinement

The different kinds of outlier pixels require different refinement strategies.

1) If p is a mismatch pixel, we search the most similar neighbor q from the

horizontal scanline w that starts from the pixel p - (N,0) to the pixel p+(N,0). And then we accept the disparity of q to p, which can be expressed as

$$D_L(p) = \{D_L(q)| \min_{q \in p \pm (N,0), q \neq p} C_{AD}(p,q)\}$$

2) If p is an occluded pixel, we first search the left and right nearest reliable disparity, respectively denoted as S_1 and S_2. If only one reliable pixel is found, $D_L(p)$ is simply replaced by $D_L(S_1)$ or $D_L(S_2)$. Otherwise, $D_L(p) = \min(D_L(s_1), D_L(s_2))$.

3.7.3 Region-based cross voting

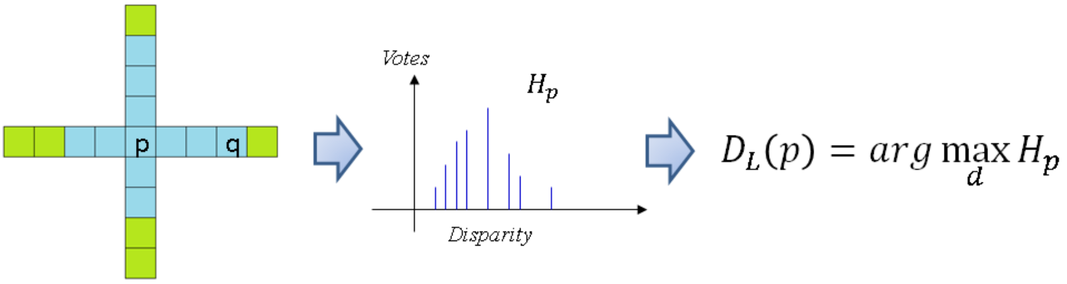


Figure 3-6 Region-based cross voting

In this step, we reuse the segment information to further refine the disparity map. For each pixel p, we build a histogram H_p with DSR+1 bins for a cross voting scheme, where DSR represents the disparity search range. The cross voting scheme is depicted in **Figure 3-6**. We survey the neighbors both from the vertical and horizontal directions, and collect the disparity votes if the neighbor q is located on the same segment of p. After votes collection, the disparity candidate with the most votes is assigned to p.

Chapter 4 Experimental Results

To evaluate a stereo algorithm, a benchmark is provided on Middlebury Website [11]. The four sequences, *Tsukuba*, *Venus*, *Teddy*, and *Cones*, are the most commonly used for testing performance. For each image pair three error measures are proposed: all image regions except for occlusions (nonocc), all regions (all), and near depth discontinuities (disc). The default error threshold is set to 1.0.

In this section we present some experimental results of the proposed method. The parameters are kept constant for all the data sets, which are presented in **Table I**.

σ_S	σ_R	$Fuse_R$	λ_m	λ_{AD}	λ_c
3	3	35	2	10	15
T_w	W_{small}	W_{big}	T_{count}	N	
100	31x31	51x51	300	15	

Table I Parameter settings for the Middlebury evaluation

4.1 Evaluation of the robust matching cost measure

First we compare the MCADSW [9] result obtained from our implement and the result after adopted the robust matching cost, which is referred as RCADSW. Both by using the winner-take-all (WTA) strategy, **Figure 4-1** shows the gain which comes from the combined matching cost measure, and the gain is independent of the aggregated window size. The average error rate can be reduced by 0.2~0.25%.

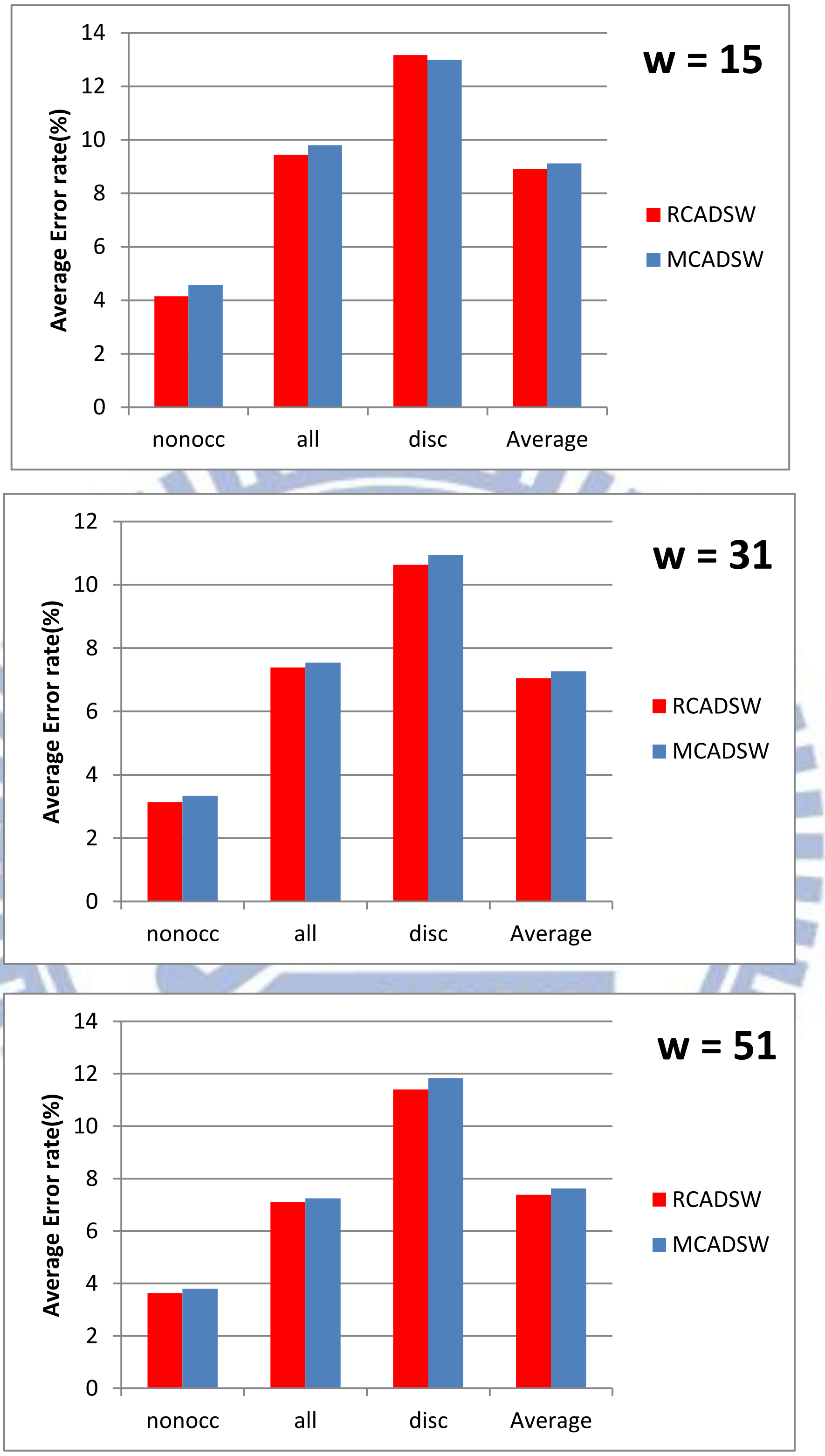
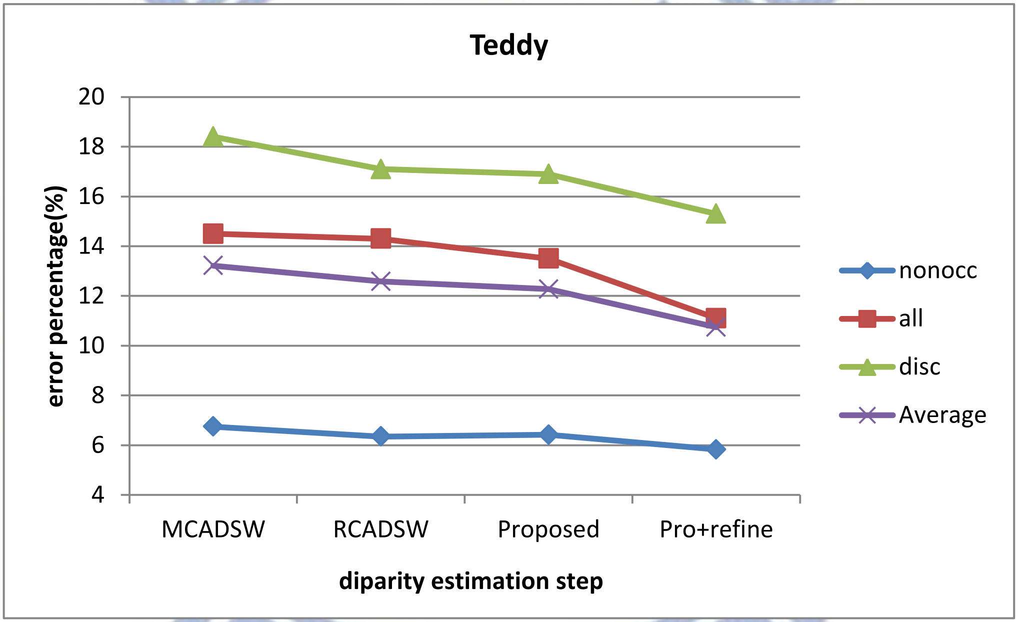
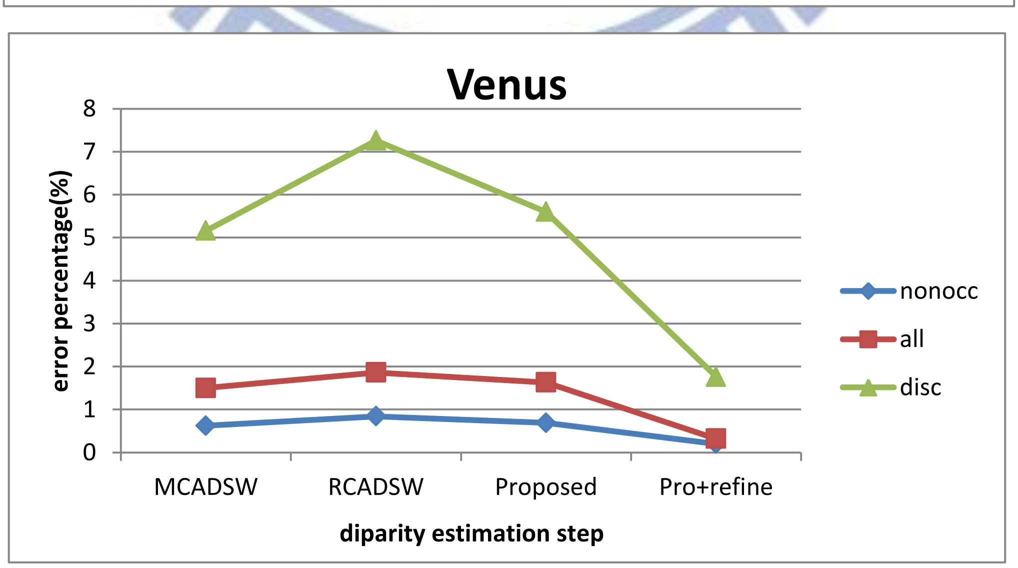


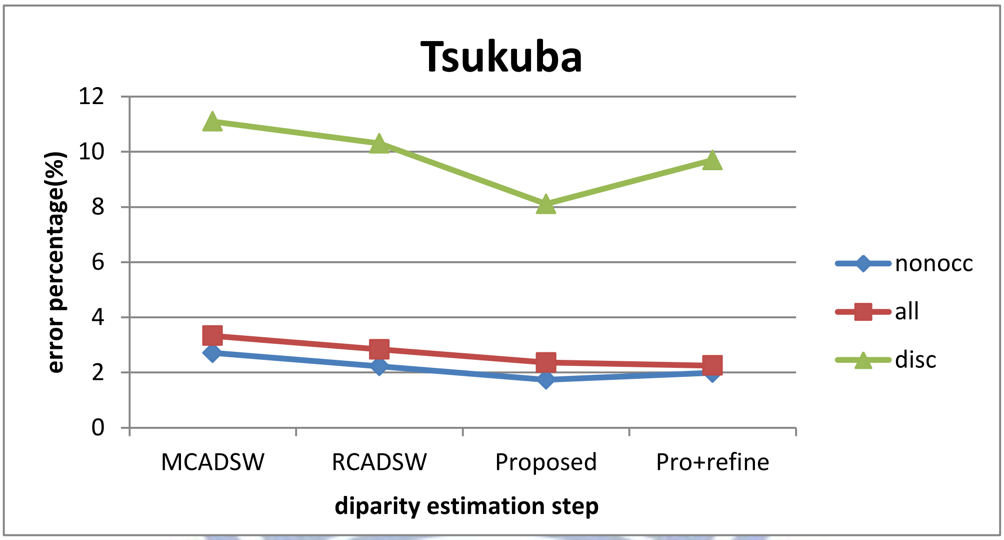
Figure 4-1 The averaged error rate in different cost measure (w means the support window size)

4.2 Compare the intermediate results of proposed method

In addition, we compared the intermediate and final disparity result of our proposed method. The support window size is fixed as 31x31 both to MCADSW and RCADSW. The results using variable window size method with and without the final refinement are respectively denoted as “Proposed” and “Pro+refine”. The quantitative evaluation of the intermediate and final disparity result is shown in **Figure 4-2**:







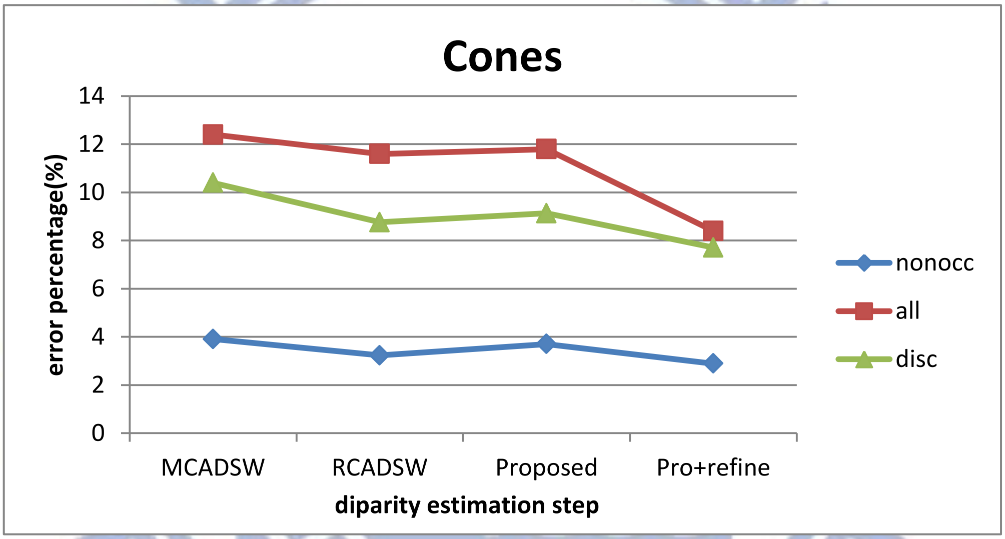


Figure 4-2 The error percentages of different error measures for 4 methods by our implementation

Table II lists the average bad pixels with average ranking of 4 methods are respectively: *Our MCADSW* 7.57% (82.3), *RCADSW* 7.10% (74.7), *proposed* 6.8% (67.3), *pro+refine* 5.63% (39.1). These experimental results show that the performance can be improved by each step in the proposed algorithm.

Algorithm	Tsukuba			Venus			Teddy			Cones			Avg.Bad Pixels
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
Our MCADSW	2.71	3.33	11.1	0.62	1.50	5.16	6.75	14.5	18.4	3.91	12.4	10.4	7.57
RCADSW	2.37	2.93	10.3	0.84	1.86	7.26	6.35	14.3	17.1	3.23	11.6	8.76	7.10
Proposed	1.74	2.36	8.11	0.69	1.63	5.60	6.42	13.5	16.9	3.70	11.8	9.13	6.80
Proposed+refine	1.99	2.25	9.70	0.20	0.32	1.76	5.83	11.1	15.3	2.89	8.40	7.71	5.63

Table II Compared the results obtained from different step in the proposed algorithm

Algorithm	Tsukuba			Venus			Teddy			Cones		
	non occ	all	disc	non occ	all	disc	non occ	all	disc	non occ	all	disc
Proposed	1.74	2.36	8.11	0.69	1.63	5.60	6.42	13.5	16.9	3.70	11.8	9.13
MCADSW [9]		2.80			0.64			13.7			10.1	
SegmentSupport [8]	2.05		7.14	1.47		10.5	10.8		21.7	5.08		12.5

Table III Comparison between proposed method and the related works using winner-take-all before refinement step

Algorithm	Tsukuba			Venus			Teddy			Cones			Avg. Bad Pixels	Avg. Rank
	non occ	all	disc	non occ	all	disc	non occ	all	disc	non occ	all	disc		
AdaptWeight [7]	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67	64.6
SegmentSupport [8]	1.25	1.62	6.68	0.25	0.64	2.59	8.43	14.2	18.2	3.77	9.87	9.77	6.44	53.7
AdaptLocalSeq [15]	1.33	1.82	7.19	0.32	0.79	4.50	5.32	11.9	14.5	2.73	9.69	7.91	5.67	44.9
Proposed	1.74	2.36	8.11	0.69	1.63	5.60	6.42	13.5	16.9	3.70	11.8	9.13	6.80	67.3
Pro+refine	1.99	2.25	9.70	0.20	0.32	1.76	5.83	11.1	15.3	2.89	8.40	7.71	5.63	39.1

Table IV Quantitative Middlebury evaluation of the propose method and the state-of-the-art local method

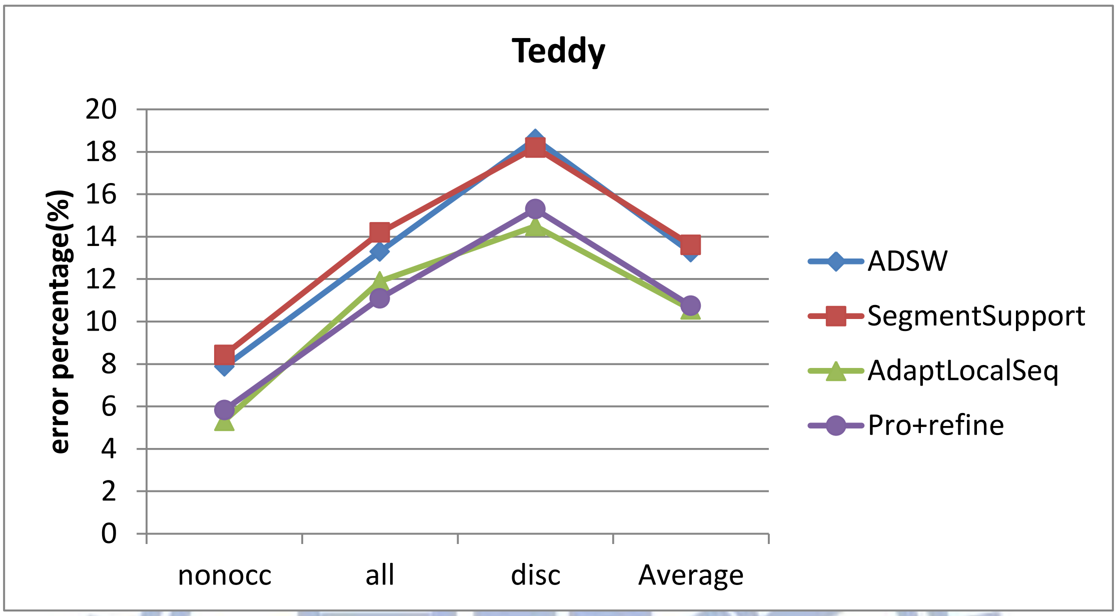
4.3 Compare with the reference works

Table III demonstrates the comparison between proposed method and the two related work on the Middlebury stereo benchmark using a winner-take-all (WTA) rule. The table also reports the results published in [8] and [9] without the refinement process which consist only part of the error measures. As it can be seen from the table, the proposed method produces a notable improvement on non-occluded and discontinuous area compared to segment support method.

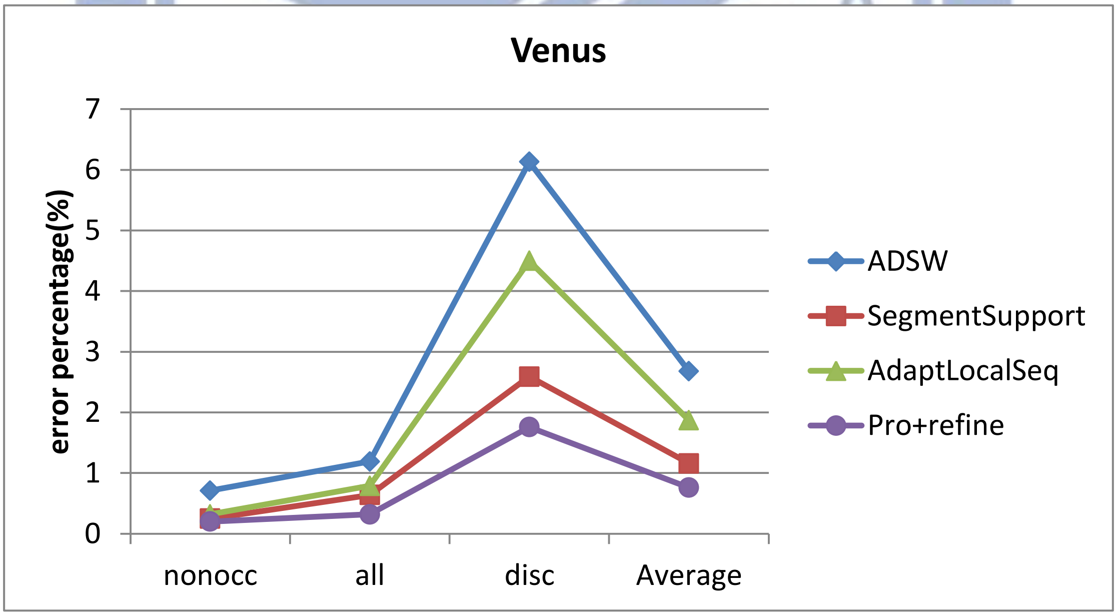
4.4 Compare with state-of-the-art methods

In the disparity refinement step, we fill the detected outliers with the reliable neighbor pixels and introduce a histogram for cross voting procedure. Then, we submit the obtained disparity maps to the Middlebury website and are compared to the current state-of-the-art local methods. The quantitative Middlebury evaluation is shown in **Table IV**. And the error percentages of different error measures for 4 test stereo pairs in shown in **Figure 4-3**.

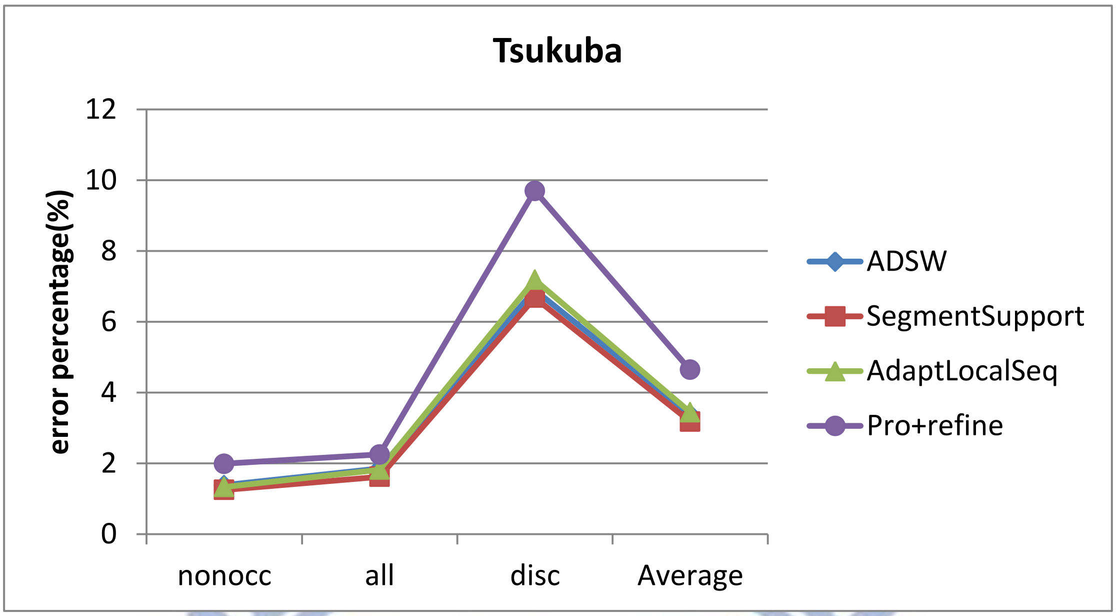
Finally, we show the disparity result obtained by the proposed method. From **Figure 4-4** to **Figure 4-7** show the disparity maps and the corresponding error maps of our proposed method and related method on the Middlebury data image sets. The error maps is evaluated by the default error threshold 1.0. The bad pixels are depicted with gray color in occluded area, otherwise the black.



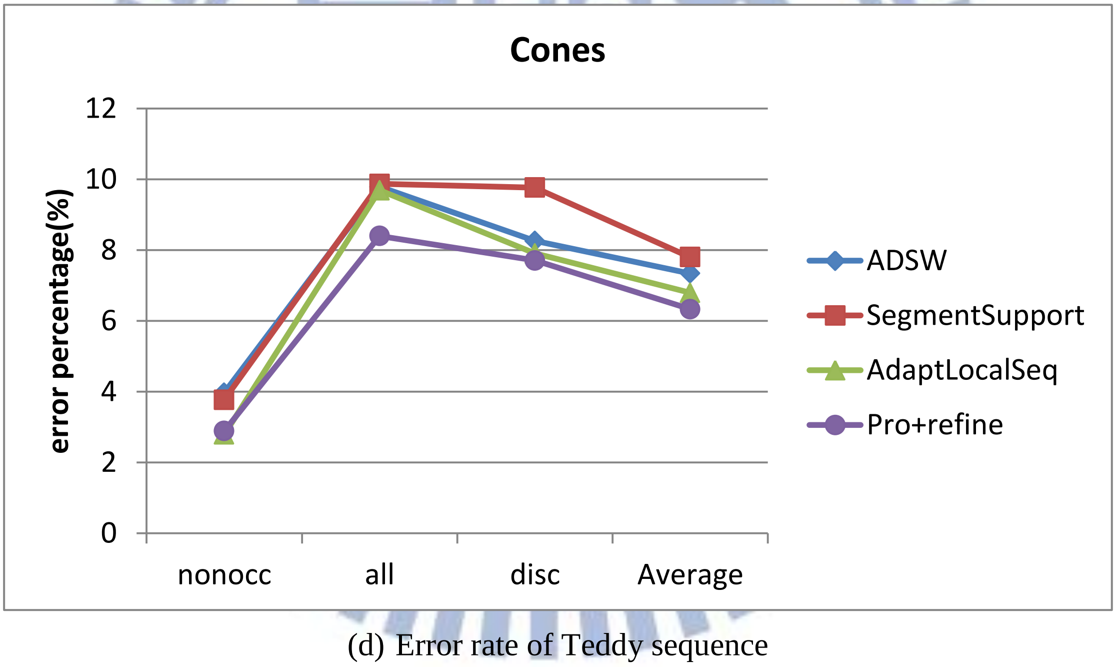
(a) Error rate of Teddy sequence



(b) Error rate of Teddy sequence

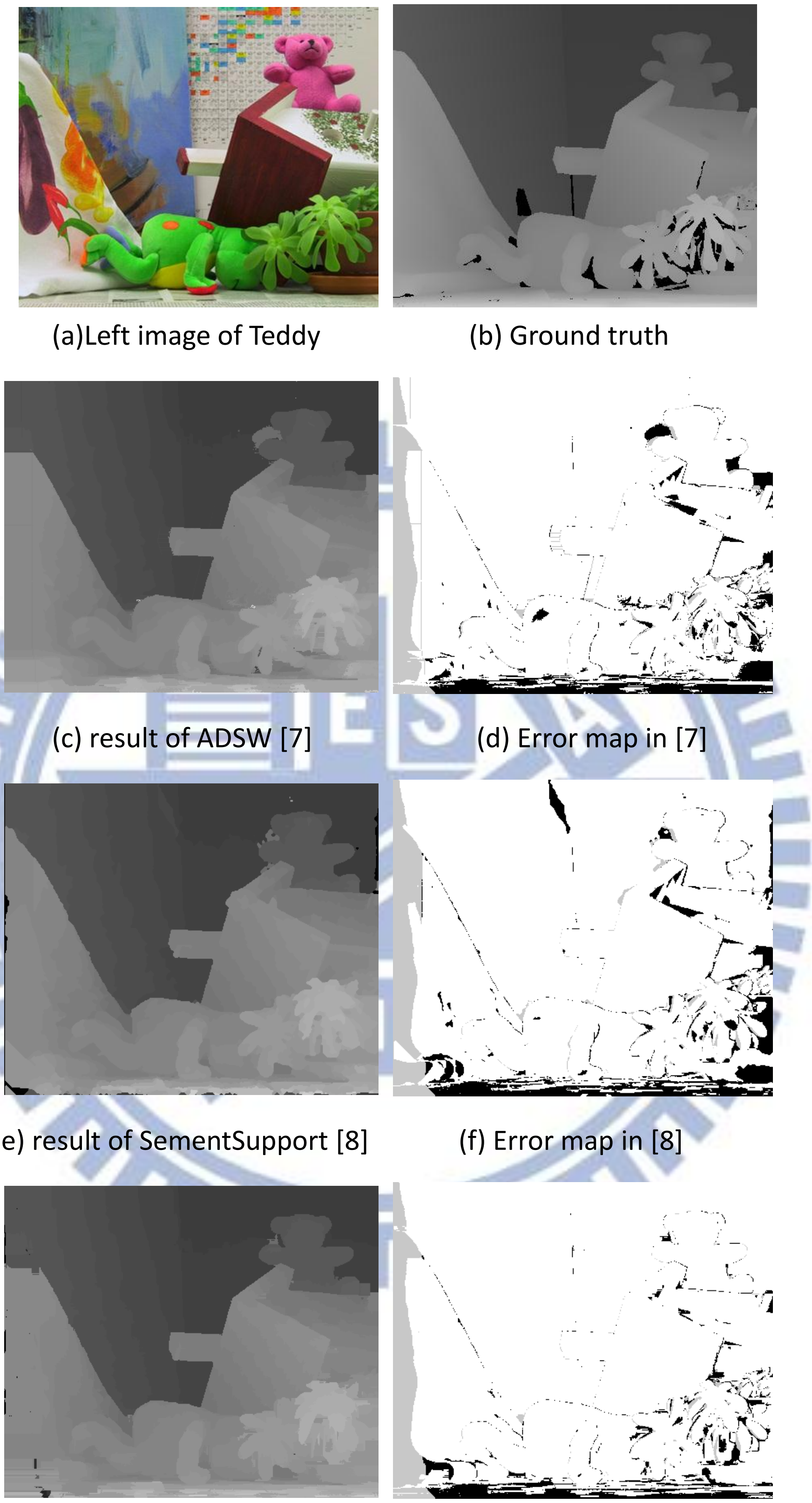


(c) Error rate of Teddy sequence



(d) Error rate of Teddy sequence

Figure 4-3 The error percentages of different error measures for 4 test stereo pairs obtained from different methods



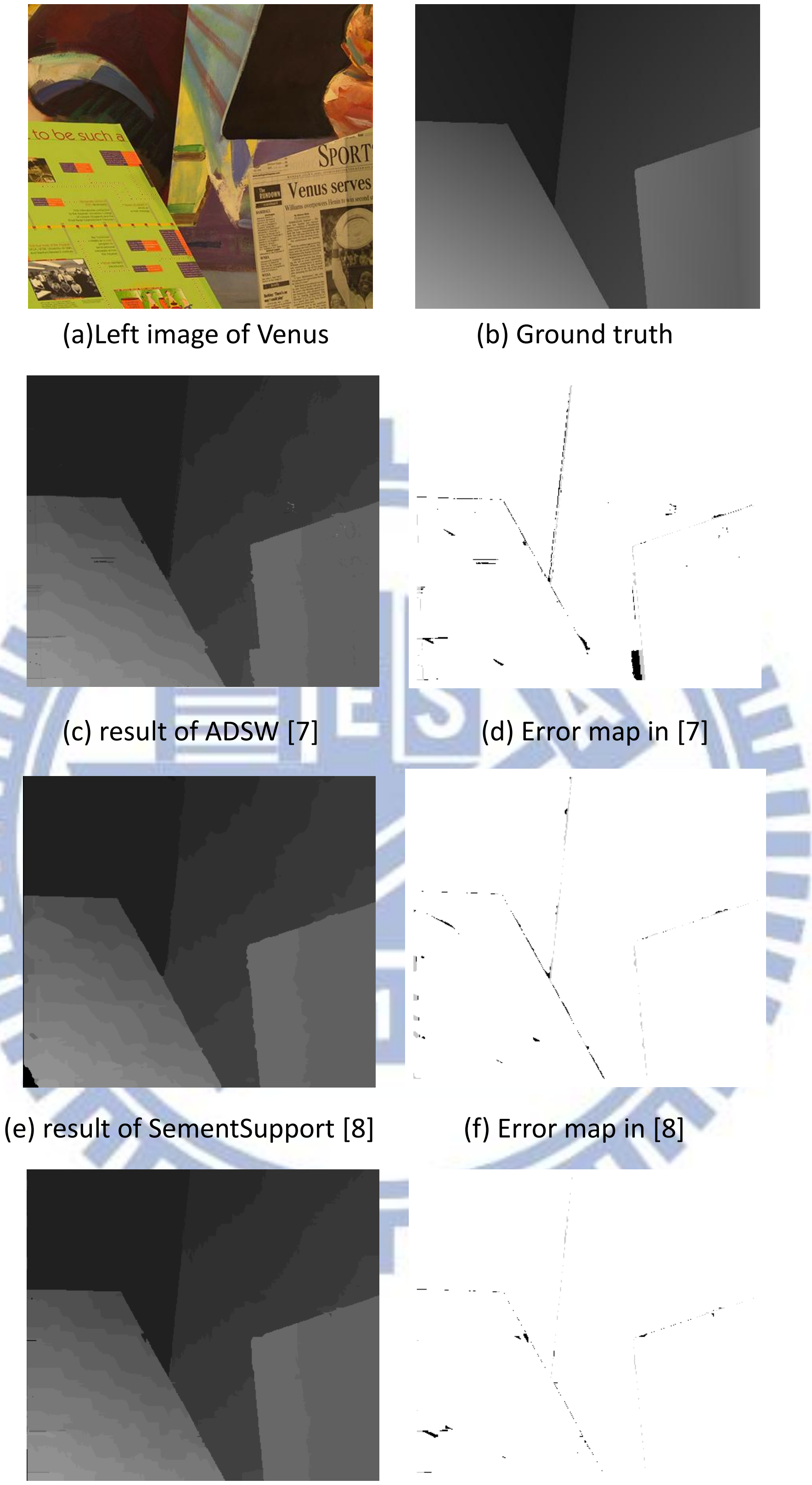
(a)Left image of Teddy (b) Ground truth

(c) result of ADSW [7] (d) Error map in [7]

(e) result of SementSupport [8] (f) Error map in [8]

(g) result of proposed method (h) Error map in proposed method

Figure 4-4 The disparity maps and the corresponding error map of our proposed method and related methods on *Teddy* sequence



(a)Left image of Venus

(b) Ground truth

(c) result of ADSW [7]

(d) Error map in [7]

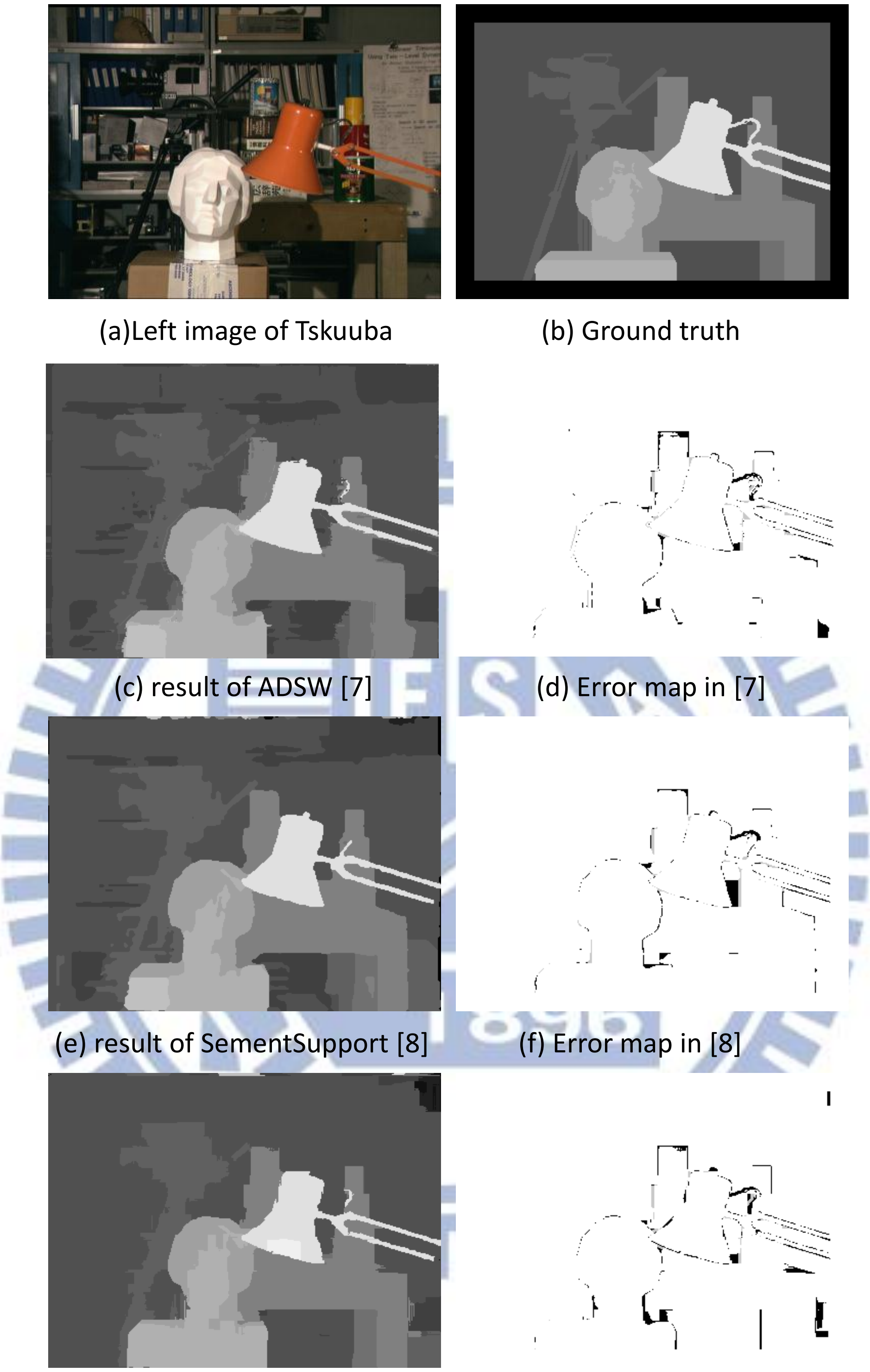
(e) result of SementSupport [8]

(f) Error map in [8]

(g) result of proposed method

(h) Error map in proposed method

Figure 4-5 The disparity maps and the corresponding error map of our proposed method and related methods on *Venus* sequence



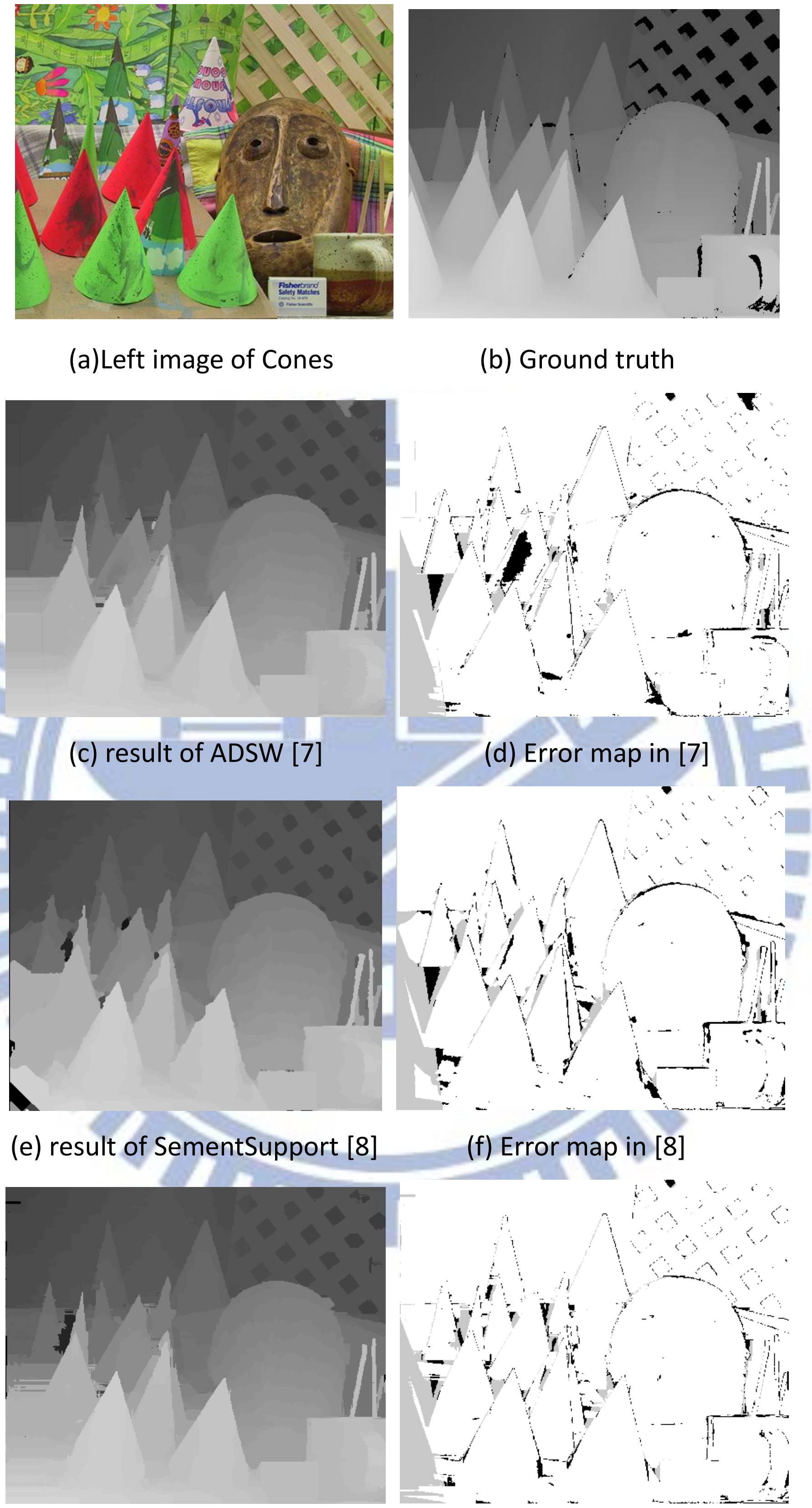
(a)Left image of Tskuuba (b) Ground truth

(c) result of ADSW [7] (d) Error map in [7]

(e) result of SementSupport [8] (f) Error map in [8]

(g) result of proposed method (h) Error map in proposed method

Figure 4-6 The disparity maps and the corresponding error map of our proposed method and related methods on *Tsukuba* sequence



(a)Left image of Cones (b) Ground truth

(c) result of ADSW [7] (d) Error map in [7]

(e) result of SementSupport [8] (f) Error map in [8]

(g) result of proposed method (h) Error map in proposed method

Figure 4-7 The disparity maps and the corresponding error map of our proposed method and related methods on *Cones* sequence

Chapter 5 Conclusion

In this paper, a combined matching cost measure with mini-census and color difference is proposed. Moreover, we propose a variable window size selection strategy before cost aggregation step. The strategy determines the support window by utilizing the segment information derived from a color based segmentation method. This information is also used for region-based cross voting scheme in refinement step. Experimental results show that the proposed method effectively improves the performance of original method in low textured region and near depth discontinuities. According the performance evaluation at the Middlebury website, the proposed method is one of the current state-of-the-art local methods.

Reference

[1] M. Tanimoto, “Free viewpoint television - FTV,” in Proceedings of Picture Coding Symposium (PCS), 2004.

[2] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Steroe Correspondence Algorithms,” International Journal of Computer Vision, vol. 47, no. 1-3, pp. 7-42, 2002.

[3] R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” in Proceeings of third European Conference on Computer Vision, vol. 2, pp. 151-158, 1994.

[4] H. Hirschmüller and D. Scharstein, “Evaluation of Stereo Matching Costs on Images with Radiometric Differences,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no.9, pp. 1582-1599, 2009.

[5] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts,” IEEE Transctions on Pattern Analysis and Machine Intellgence, vol. 23, no. 11, November 2001.

[6] P. fua, “Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities,” 12th International Joint Comference on Artificial Intelligence, pp. 1292-1298, 1993.

[7] K.-J. Yoon and I.-S. Kweon, “Adaptive Support-weight Approach for Correspondence search,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, No.4, pp. 650-656, April 2006.

[8] F. Tombari, S. Mattoccia, and L. Di Stefano, “Segmentation-Based Adaptive Support for Accurate Stereo Correspondence,” Lecture Notes in Computer Science, vol. 4872, pp. 427-438, 2007.

[9] 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 Support Weight,” IEEE Transactions on Circuit and System for Video Technology, vol. 20, no. 6, June 2010.

[10] Z.-F. Wang and Z.-G. Zheng, “A Region Based Stereo Matching Algorithm Using Cooperative Optimization,” IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1-8, June 2008.

[11] D. Scharstein and R. Szeliski, “Middlebury stereo evaluation - version 2,” http://vision.middlebury.edu/stereo/eval/, 2010.

[12] D. Comanicu, P. Meer, “Mean Shift: A Robust Approch Toward Feature Space Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002.

[13] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an Accurate Stereo Matching System on Graphics Hardware," Computer Vision Workshops, pp. 467-474, 2011.

[14] H. Hirschmüller, "Stereo Processing by Semiglobal Matching and Mutual Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 328-341, Feb. 2008.

[15] E.T. Psota, J. Kowalczuk, J. Carlson, and L. C. Perez, "A Local Iterative Refinement Method for Adaptive Stereo Matching," IPCV, 2011.

[16] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and Evaluation of Cost Aggregation Methods for Stereo Correspondence," in Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 24-26, June 2008.