

Stereo matching algorithm based on per pixel difference adjustment, iterative guided filter and graph segmentation[☆]

Rostam Affendi Hamzah ^{a,b,*}, Haidi Ibrahim ^a, Anwar Hasni Abu Hassan ^a

^a School of Electrical & Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

^b Department of Electronic and Computer Engineering Technology, Faculty of Engineering Technology, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia



ARTICLE INFO

Article history:

Received 22 April 2016

Revised 19 November 2016

Accepted 25 November 2016

Available online 26 November 2016

Keywords:

Iterative guided filter

Disparity map

Gradient difference

Absolute difference

Undirected graph segmentation

Stereo matching algorithm

ABSTRACT

Stereo matching process is a difficult and challenging task due to many uncontrollable factors that affect the results. These factors include the radiometric variations and illumination inconsistency. The absolute differences (AD) algorithms work fast, but they are too sensitive to noise and low textured areas. Therefore, this paper proposes an improved algorithm to overcome these limitations. First, the proposed algorithm utilizes per-pixel difference adjustment for AD and gradient matching to reduce the radiometric distortions. Then, both differences are combined with census transform to reduce the effect of illumination variations. Second, a new approach of iterative guided filter is introduced at cost aggregation to preserve and improve the object boundaries. The undirected graph segmentation is used at the last stage in order to smoothen the low textured areas. The experimental results on the standard indoor and outdoor datasets show that the proposed algorithm produces smooth disparity maps and accurate results.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

In recent years there have been various progresses in the field of image processing and computer vision. One of the popular topics in computer vision is the generation of depth map or disparity map from a pair of stereo images. This topic is useful and plays an essential role in many applications such as three dimensional (3D) scanning, 3D tracking, 3D reconstruction and autonomous navigation. Many research groups have studied this discipline in depth, gaining mechanism for 3D mapping. Following this, stereo vision is established with focusing on achieving low computational cost and high accuracy. It uses two parallel digital cameras to acquire the depth of a scene. The stereo cameras provide high resolution images at low prices and every pixel of the images can be used for any other applications as well [1].

The matching process of stereo matching searches for corresponding predictions of the same scene point onto both camera planes. One of the problems associated with developing image matching is the computational cost required to achieve the appropriate results [1]. The result from the matching process is presented as a disparity map. This map provides the depth data

which is important to 3D image reconstruction. The disparity map estimation comprises of finding the correspondence for each pixel pair from two images at a designated coordinates (i.e., reference image coordinates). Most of the stereo matching algorithms rely on four steps taxonomy proposed by Scharstein and Szeliski [2]. These steps are:

- Step 1: Matching cost computation (i.e., matching process for each pixel from left image to right image).
- Step 2: Cost aggregation (i.e., aggregate initial costs over a support region).
- Step 3: Disparity optimization (i.e., select the disparity level that optimize the function).
- Step 4: Disparity refinement (i.e., post-processing to refine the final disparity map).

The disparity map algorithms can be classified as either global, semi-global or local methods. This classification depends on how the disparity is been calculated [2].

Global optimization methods treat the disparity assignment problem as a problem of minimizing a predefined global energy function. They are usually less sensitive to local individualities and tend to be more computationally expensive. The measurement is taken from the global data with an additional smooth constraint for neighbouring pixels [3]. Numerous methods for solving the global energy minimization problem by using a graph-based from

* This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author at: School of Electrical & Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia.

E-mail address: rostamaffendi@utem.edu.my (R.A. Hamzah).

Markov Random Field (MRF) have been proposed [4]. These methods can be categorized as either graph cut (GC) method [5] or belief propagation method [6].

The GC method gives the minimal energy solution by applying minimum cut and max-flow algorithms to the energy flow structure which is extracted from the MRF graph. In contrast, the belief propagation method minimizes the energy function by iteratively passing messages from the current node to neighboring nodes in the MRF graph. Mozerov et al. [4] utilized two energy minimization steps by using bilateral filter (BF) on a fully attached MRF. First, the minimum energy marginally to the globally connected model are calculated. These values are used for the second step of optimization with locally connected model. Their algorithm was capable to reduce the occlusion errors and increase the efficiency but the computational complexity with two steps of optimization also increase tremendously.

Semi-global matching (SGM) is another method to find the correspondence pixels between a pair of stereo images. This method uses local approximation to form matching cost and aggregates using a global cost function across entire 2D MRF image along linear 1D pixel paths [7]. These paths may use several directions through the images (e.g., 8 or 16 directions) to cover the structure of the image. However, the SGM depends on both the number of pixels and the disparity range. It makes the SGM to require more space of temporary memory to be operated. Wenzel et al. [8] implemented SGM method which performed multi-baseline matching between a base image with all matched images.

The modification of SGM by Bethmann et al. [9] produced best matches in 3D space. This method transferred the process of cost calculation and aggregates from the image into the object space. Sinha et al. [10] proposed an algorithm using sparse feature matching followed by an iterative clustering steps. Local plane sweeps are then executed around each slanted plane to create out-of-plane parallax and matching cost estimations. Each pixel is assigned to one of the local plane hypotheses by using the SGM optimization. This technique delivers significant speedups for high resolution images but the algorithm still produces low accuracy on the low textured regions.

On the other hand, local methods determine the disparity using the correspondence between the gray values or patterns within a given local support window. The local support window is a small number of pixels around the pixel of interest. These methods are also referred as window-based or area-based methods. There are several approaches related to window-based methods such as fixed window [11], multiple window [12], adaptive window [13–15] and segmentation based [16]. Such approaches use only local information. They have low computational requirement and a short runtime. The disparity map is determined by selecting the smallest matching cost value from the disparity candidates. This selection is well known as winner take all (WTA) approach. Although the local methods can produce disparity maps quickly, thus precision is low, especially at regions with low textured and depth discontinuity. These problems can be reduced using the image segmentation process such as mean shift [3] or watershed segmentations [17].

One of the well-known local-based methods is using adaptive support weight (ASW) which was proposed by Yoon and Kweon [18]. The ASW is an adaptive method which each pixel has a different support weight. Hosni et al. [19] proposed the ASW using a guided filter (GF) which was developed by He et al. [20,21] to reduce the edge flattening problem at the cost aggregation stage. They performed color segmentation inside the GF's window. The pixels having a similar color with the center pixel are imposed with a high support weight. Zhang et al. [22] used exponential function with different window sizes of GF at cost aggregation stage. They applied the ASW technique to reduce the discontinuity error. Yang

et al. [23] used two support windows of the GF to increase the edge-preserving efficiency, but this technique also increases the weight of invalid pixels. Zhu et al. [24] used the ASW using a GF with modified dynamic programming. Their results show the reduction of errors on the occluded and discontinuity regions. However this method is unable to reduce the errors especially on the low textured or flattening areas. Zhu et al. [25] used adaptive edge-preserving GF with cross-based support window to resolve the ambiguity pixels effectively. Kordelas et al. [26] proposed a new approach which used content-based GF and weighted SGM for accurate disparity map estimation. They used two different sizes of support windows which a pixel was assigned based on the local image content around the pixel of interest. Their results demonstrated high accuracy based on the Middlebury database V2.

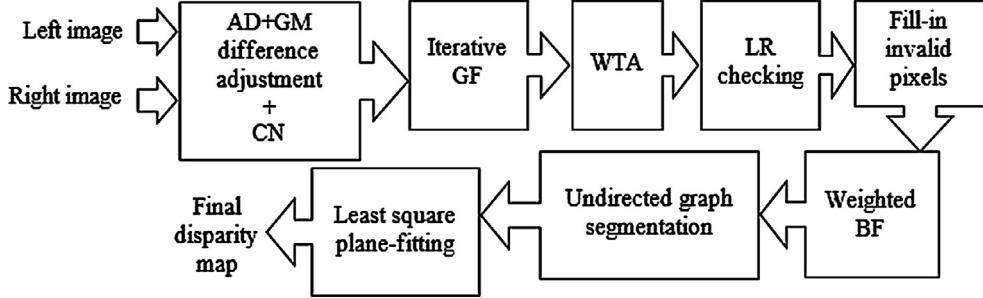
Based on the studies from state of art algorithms, every method (i.e., global, semi-global, local) have their own advantages and disadvantages [27]. Global methods commonly deliver more accurate disparity map [28]. However, they are time-consuming and high computational complexity. So they are relatively slow and do not scale well to high resolution images that involve high storage. The SGM methods show more fast execution and memory efficient compared to the global methods. These methods aim to minimize a global 2D energy function by solving a large number of 1D minimization problem [29]. Generally, the state of art local methods lack of accuracy but provide high speed and low computational complexity. However based on the recent studies, this method is able to produce high accuracy. This is shown by the works of Kordelas et al. [14], Zhan et al. [30] and Peng et al. [31] which produce high accuracy and fast execution. Their algorithms also rank at top of Middlebury stereo evaluation based on the lowest of bad pixels percentage [32]. Encouraged by this, the proposed work in this manuscript is using local approach.

This paper proposes a new local-based stereo matching algorithm to compute the disparity maps more accurately. The proposed work uses a combination of three different similarity measures which consists of absolute differences (AD), gradient matching (GM) and census transform (CN). The differences of AD and GM are imposed with per-pixel difference adjustment to increase the quality of the preliminary disparity map. The aggregation step uses a new developed iterative GF which increases the accuracy of the edges and discontinuity areas. The final stage consists of post-processing steps with the usage of weighted BF, undirected graph segmentation and plane fitting process.

The rest of this paper is organized as follows. The next section will provide a detailed explanation of the proposed stereo matching algorithm. This is followed by the section of experimental arrangements and the results. The conclusion is provided in the last section.

2. The proposed stereo matching algorithm

Similar to a basic local stereo matching development, the proposed algorithm involves the steps as explained in Section 1. The block diagram of the proposed algorithm is shown by Fig. 1. The new matching cost computation is using a combination of three per-pixel difference measurements with adjustment element. The cost aggregation is implemented using the iterative GF. Then, the optimization step uses a winner-take-all (WTA) strategy. The WTA strategy absorbs the minimal aggregated corresponding value for each valid pixel. At this stage, some invalid or unwanted pixels are still occurred at the occlusion and untextured areas. These unwanted pixels will be detected by left-right (LR) consistency checking process [26,33]. Then, the fill-in process is carried out to replace the invalid pixels with a valid minimum pixel value. The disparity refinement step consists of implementing the

**Fig. 1.** Block diagram of the proposed algorithm.

weighted bilateral filter to remove the remaining noise which usually occurs during the fill-in process. The undirected graph segmentation and least square plane fitting process are used at the final stage to recover the low texture regions on the disparity map.

2.1. Matching cost computation

The state of art AD algorithm as implemented by Tan and Monasse [34] is exposed to high distortions. It increases the errors on the disparity map especially at the object boundaries and low textured areas. To solve this problem, the work presented in this paper proposes an algorithm which is able to reduce and minimize the errors. The new proposed algorithm in this work has introduced β at the difference of AD and GM. The $AD(p, d)$, which relies only on the intensity difference between two pixels in RGB channels at the left image I_l and right image I_r as implemented in [34] given by Eq. (1):

$$AD(p, d) = \sum_{i \in \{R,G,B\}} |I_l^i(p) - I_r^i(p - d)| \quad (1)$$

where the pixel of interest coordinates (x, y) represented by p , i denotes the RGB channels number and d is disparity value. The new proposed $AD_{new}(p, d)$ algorithm is given by Eq. (2):

$$AD_{new}(p, d) = \beta AD(p, d) \quad (2)$$

where β indicates an element of a constant value to adjust the intensity differences. The condition while applying the final absolute differences $AD'(p, d)$ is given by Eq. (3):

$$AD'(p, d) = \begin{cases} \tau_{AD}, & \text{if } AD_{new}(p, d) > \tau_{AD}, \\ AD_{new}(p, d), & \text{otherwise.} \end{cases} \quad (3)$$

where τ_{AD} denotes the truncated value as implemented by Ploumpis et al. [35] to increase the robustness against the outliers.

Meanwhile, to calculate the components of the gradient from each image, the gradient value in horizontal direction G_x and vertical direction G_y are being used as implemented by von Gioi et al. [36]. Fundamentally, the gradient values are given by Eqs. (4) and (5):

$$G_x = [1 \ 0 \ -1] * I \quad (4)$$

$$G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * I \quad (5)$$

where I is the image and $*$ is the convolution operation. Using both gradient components, the gradient magnitude m is given by Eq. (6):

$$m = \sqrt{G_x^2 + G_y^2} \quad (6)$$

The modulus of gradient operator in Eq. (6) is applied to the left gray image m_l and right gray image m_r respectively. In the x -

direction of gradient displacement with a static position of y -direction, the gradient matching cost $GM(p, d)$ is given by Eq. (7):

$$GM(p, d) = |m_l(p) - m_r(p - d)| \quad (7)$$

where the pixel of interest's coordinates is p and d denotes the disparity value. The gradient difference in Eq. (8) is also imposed with β element with the same value as used in Eq. (2):

$$GM_{new}(p, d) = \beta GM(p, d) \quad (8)$$

The final value of the gradient differences is given by Eq. (9):

$$GM'(p, d) = \begin{cases} \tau_{GM}, & \text{if } GM_{new}(p, d) > \tau_{GM}, \\ GM_{new}(p, d), & \text{otherwise.} \end{cases} \quad (9)$$

where τ_{GM} denotes the threshold or truncated value of GM. The matching cost function $M'(p, d)$ at this stage is the combination of $AD(p, d)$ and $GM'(p, d)$ which is given by Eq. (10):

$$M'(p, d) = \alpha AD'(p, d) + (1 - \alpha) GM'(p, d) \quad (10)$$

where the α is added to balance the color and gradient terms such implemented by Yang et al. [23]. The α controls the sensitivity to radiometric differences.

In this work, the matching cost is added with the census transform volume. This makes the final matching cost volume robust against the illumination changes. The census transform uses local information which computes the Hamming distance of bit strings given by Eq. (11):

$$CN(p) = \otimes_{q \in w_{CN}} cen(p, q) \quad (11)$$

where \otimes operator refers to a bit-wise catenation, p and q denote the pixel of interest and neighbouring pixels respectively. The w_{CN} is a support window and $cen(p, q)$ represents the binary function with the conditions as given by Eq. (12):

$$cen(p, q) = \begin{cases} 1, & I(p) \geqslant I(q), \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

where $I(p)$ and $I(q)$ are the pixel of interest and neighbouring pixel values respectively. The Hamming distance is used to calculate the difference between two bit strings (i.e., left and right images) which is given by Eq. (13):

$$CN'(p, d) = \text{Hamming}(CN_l(p) - CN_r(p - d)) \quad (13)$$

where CN_l and CN_r are the bit strings of left and right images respectively. The final matching cost function $M(p, d)$ is given by Eq. (14) which uses normalised cost function as implemented by Lee et al. [37].

$$M(p, d) = 2 - \exp(-M'(p, d)) - \exp(-CN'(p, d)) \quad (14)$$

2.2. Cost aggregation

Cost aggregation step is the most important stage which minimizes the matching uncertainties. It produces overall performance of the disparity maps for local methods. In this work, the iterative GF is proposed to reduce the noise and preserve the edges. This method will improve the edge properties and increase the efficiency. The guided means the filter is using the selected guided imaging (i.e., left or right grayscale image as a guide for the filtering process). The left image is selected in this work as a reference and the guidance to the process of filtering. The filter kernel of the iterative GF $G_{p,q}(I_n)$ is given by Eq. (15):

$$G_{p,q}(I_n) = \frac{1}{|w|^2} \sum_{q \in w_k} \left(1 + \frac{(I_{p,n-1} - \mu_{k,n-1})(I_{q,n-1} - \mu_{k,n-1})}{\sigma_{k,n-1}^2 + \epsilon} \right) \quad (15)$$

where I_n represents the guidance grayscale image at n -th iterations and the coordinates for pixel of interest (x, y) is denoted by p . The w_k is the support window with the size of $r \times r$ pixels. The w refers to the number of pixels in w_k . The neighbouring pixels in the support window represented by q and k denotes as a center pixel. The σ and μ are the variance and mean of the intensity values on the guidance image. The ϵ represents the control element for the smoothness term. This work uses the same approach in [21] for the guidance grayscale image during the iteration. If $n = 0$, there is no filtering activity and if $n = 1$ and above, the image will be filtered accordingly. The aggregation cost volume $CA(p, d)$ at this stage is given by Eq. (16):

$$CA(p, d) = G_{p,q}(I_n)M(p, d) \quad (16)$$

where $G_{p,q}(I_n)$ is the weight of iterative GF and $M(p, d)$ represents the input filtering image. The weight of the edge preserving factor is determined by the sum of neighbouring pixels of q in Eq. (15) within the support region of the guidance image.

2.3. Disparity optimization

To obtain the accurate disparity map, this work computes the final disparity by selecting the minimal aggregated corresponding value for each pixel using WTA strategy. The utilization of WTA strategy for local algorithms are able to reduce the computational complexity as implemented in [38,39]. However, through their findings, the disparity maps attained at this stage still having errors in the unmatched pixels or occluded regions. Based on the Eq. (17), the disparity associated with the minimum aggregated cost d at each pixel is chosen. The $CA(p, d)$ means the cost aggregation volume which acquired from the process of cost aggregation and d_r represents the set of allowed discrete disparity values in an image.

$$d = \arg \min_{d \in d_r} CA(p, d) \quad (17)$$

2.4. Disparity refinement

The final stage of the proposed algorithm consists of post-processing steps. It comprises of several consecutive processes which are invalid disparity detection, fill-in invalid disparity value, filtering, image segmentation [40] and plane-fitting process. The invalid disparity locations are detected by LR consistency checking process which are produced by occluded and some flat regions of a scene. This process is performed from left reference disparity map image that coincides with the right reference disparity map. The inconsistent values between the two maps are assigned as invalid disparity. Therefore, using the same approach used by Kordelas et al. [26] and Mattoccia et al. [33] to detect the outliers, the disparity values do not satisfy the relation of Eq. (18):

$$|d_{LR}(p) - d_{RL}(p - d_{LR}(p))| \leq \tau_{LR} \quad (18)$$

where d_{LR} denotes the left reference and d_{RL} is the right reference of disparity maps. The τ_{LR} represents the threshold value which is $\tau_{LR}=0$ for outliers detection and p is the position of the targeted disparity.

Then, the fill-in invalid disparity process takes place. In this work, the left image is a reference and the filling process starts with left to the right valid pixel replacement. The invalid disparity is replaced with a nearest valid disparity value. The valid value must be located at the same scan line or at the starting scan line as shown by Eq. (19):

$$d(p) = \begin{cases} d(p-i), & d(p-i) \leq d(p+j), \\ d(p+j), & \text{otherwise.} \end{cases} \quad (19)$$

where $d(p)$ is a disparity value at the location of p , $(p-i)$ represents the location of the first valid disparity on the left side and $(p+j)$ represents the location of the first valid disparity on the right side. However, this filling and replacing process produces the unwanted streak artefacts on the disparity map. To remove the noise, the weighted bilateral filter $B(p, q)$ is utilized as given by Eq. (20):

$$B(p, q) = \exp \left(-\frac{|p-q|^2}{\sigma_s^2} \right) \exp \left(-\frac{|d(p) - d(q)|^2}{\sigma_c^2} \right) \quad (20)$$

where p is the disparity location needs to be denoised using weight of the neighbouring values. The σ_s represents a spatial adjustment parameter and σ_c corresponds to the disparity similarity parameter. The σ value of zero in this filter indicate a spatial or disparity similarity domain of zero, meaning no filtering is being executed [41]. The $|p-q|$ refer to spatial Euclidean distance and $|d(p) - d(q)|$ is the Euclidean distance in disparity value. Bilateral filter is an edge-preserving filter and is able to improve the disparity maps quality. This filter applies a higher weight to disparities that spatially close and having a similar value according to the sigma adjustment [34]. The weighted median filter was implemented in [42,43]. Their approaches achieved high accuracy on noise removing. According to this, from Eq. (20), the weighted of $B(p, q)$ is changed to a summation of histogram $h(p, d_r)$ which produces Eq. (21):

$$h(p, d_r) = \sum_{q \in w_p | d(q) == d_r} B(p, q) \quad (21)$$

where d_r denotes the disparity range (i.e., $d_r \in [\text{disparity range}]$) and w_p is the window size with the radius $r \times r$ at centred pixel of p . The final disparity value d' is determined by the median value of $h(p, d_r)$ given by Eq. (22):

$$d' = \text{med}\{d | h(p, d_r)\} \quad (22)$$

This work employs undirected graph segmentation and least square plane-fitting techniques to get the final disparity maps. The undirected segmentation process is based on two steps. This segmentation process is the type of an edge and it completely checks the disparity map. First, the edges are selected through the weights on each edge of dissimilarity measure between the disparity values. Let edge E contains vertices v_p and v_q , (i.e., $(v_p, v_q) \in E$ corresponds to a pair of disparity values). Each $(v_p, v_q) \in E$ has a weight of dissimilarity given by Eq. (23):

$$\omega_{\text{seg}}(v_p, v_q) = |d'(p) - d'(q)| \quad (23)$$

where $\omega_{\text{seg}}(v_p, v_q)$ is a weight of dissimilarity between the elements of v_p , and v_q . The $d'(p)$ and $d'(q)$ are the target and neighbouring disparity values. Any edge components in E are segmented from the conditions given as follows. The edges between two disparity values in the same component should have relatively low weights and edges between pixels in different components should have higher weights. These two groups will form the segmented region

of $C \in S$ where C is a component and S is a segment. Second, this method evaluates the boundary between the two components to determine the segmentation group within these two regions which is given by Eq. (24):

$$\Delta(C_1, C_2) = \min_{v_p \in C_1, v_q \in C_2, (v_p, v_q \in E)} \omega_{\text{seg}}(v_p, v_q) \quad (24)$$

where $\Delta(C_1, C_2)$ is the difference between the two components from the minimum value of $\omega_{\text{seg}}(v_p, v_q)$. The final segmentation process $S(C_1, C_2)$ is given by Eq. (25):

$$S(C_1, C_2) = \begin{cases} \text{true}, & \text{if } \Delta(C_1, C_2) < \Delta_{\min}(C_1, C_2), \\ \text{false}, & \text{otherwise.} \end{cases} \quad (25)$$

where C_1 and C_2 will be merged as a region if the condition is *true*, otherwise the C_1 and C_2 may remain as a different segment. The minimum internal difference $\Delta_{\min}(C_1, C_2)$ is given by Eq. (26):

$$\Delta_{\min}(C_1, C_2) = \min(\text{Int}(C_1) + \tau_{\text{seg}}(C_1), \text{Int}(C_2) + \tau_{\text{seg}}(C_2)) \quad (26)$$

where $\text{Int}(C)$ is the internal difference weight value in a component given by Eq. (27):

$$\text{Int}(C) = \max_{e \in MST(C, E)} w(e) \quad (27)$$

where $\text{Int}(C)$ uses the maximum weight $w(e)$ in the minimum spanning tree of a component $MST(C, E)$. The τ_{seg} is a threshold function given by Eq. (28):

$$\tau_{\text{seg}}(C) = \frac{k}{|Z_c|} \quad (28)$$

which k is a constant parameter and Z_c is the size of a component. The constant parameter sets a scale of observation, whereby a large scale causes for larger components in size.

The plane fitting process is executed as implemented by [44]. Each segmented region from Eq. (25) must have a constant disparity value and will have the same disparities in the disparity plane which is modeled by Eq. (29):

$$d^k(x, y) = ax + by + c \quad (29)$$

where $d^k(x, y)$ and (x, y) represent the disparity plane and the coordinates of the k^{th} segment. The a and b are the slope parameters and c represents the distance of the plane to the origin. These parameters can be determined based on the least squares calculations by Eq. (30):

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i d_i^k \\ \sum_{i=1}^m y_i d_i^k \\ \sum_{i=1}^m d_i^k \end{bmatrix} \begin{bmatrix} \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix}^{-1} \quad (30)$$

where (x_i, y_i) and m denote the coordinates and number of pixels in k^{th} segment. The d_i^k is the disparity value in the k^{th} segment. The final approximated disparity plane is using the same parameters

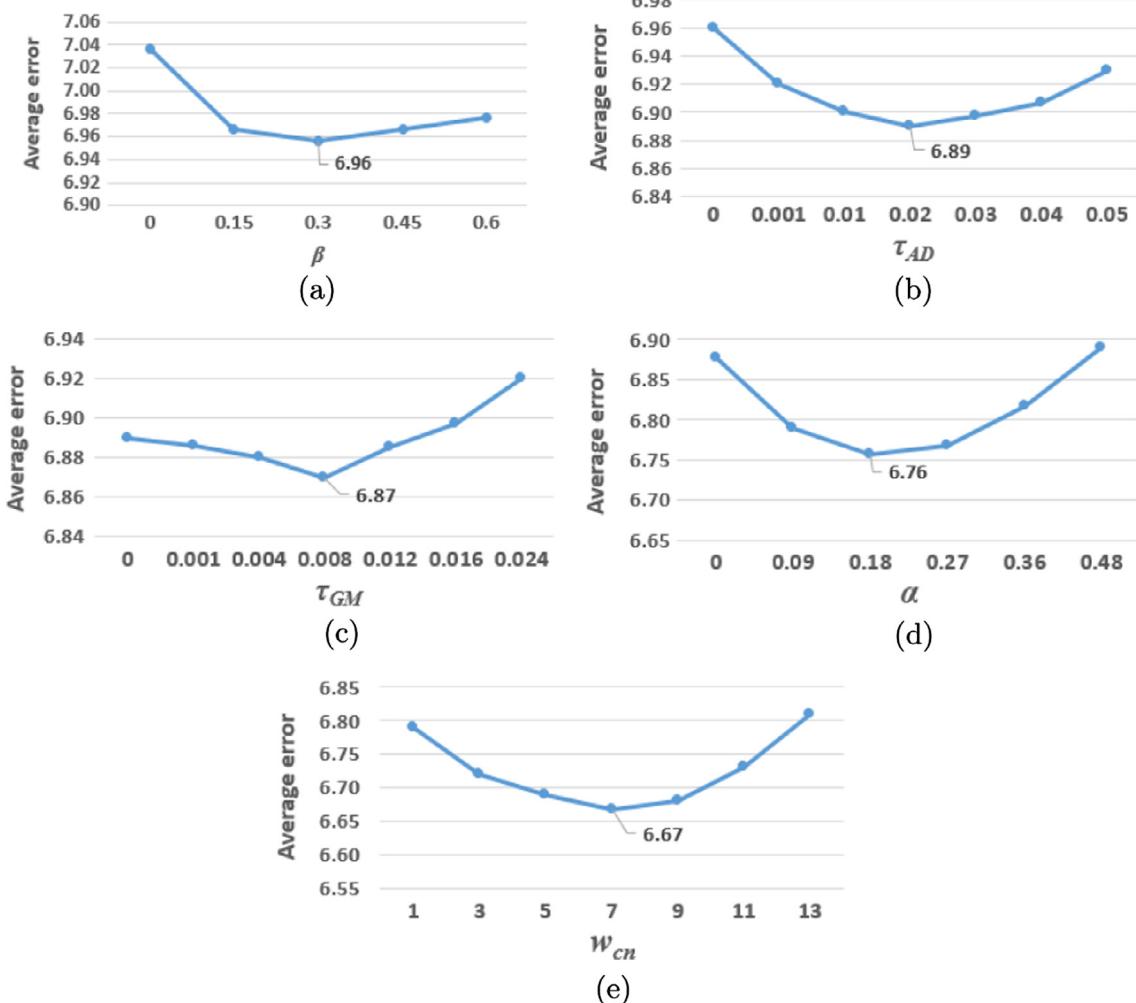


Fig. 2. The experimental results on parameter settings at Step 1 using the Middlebury training dataset. The parameters are (a) β (b) τ_{AD} (c) τ_{GM} (d) α and (e) w_{cn} .

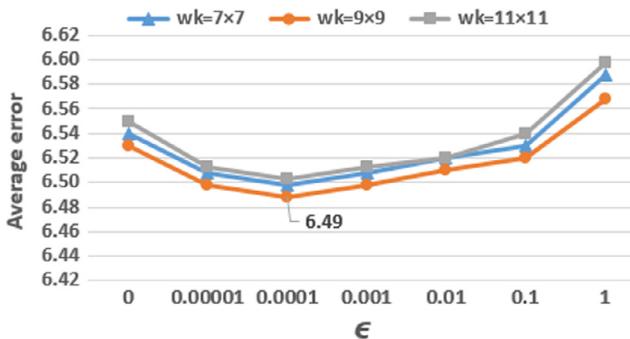


Fig. 3. The experimental result of iterative GF parameters (i.e., w_k and ϵ) at cost aggregation step.

in Eqs. (29) and (30) to obtain the disparity values. The planar image segment is considered acceptable when the condition of the threshold value is not exceeded which is given by Eq. (31):

$$C(S, P) = \begin{cases} \text{true}, & \text{if } \delta \leq \tau_{\text{plane}}, \\ \text{false}, & \text{otherwise.} \end{cases} \quad (31)$$

where S is the segment, P represents the disparity plane, τ_{plane} is the plane threshold value and δ is the average distance given by Eq. (32):

$$\delta = \frac{1}{m} \sum_{i=1}^m |d_i^k - (ax_i + by_i + c)| \quad (32)$$

where d_i^k is the disparity in a segment and $(ax_i + by_i + c)$ denotes the disparity plane value.

n	Adirondack	zoomed image	Error	
			nonocc (%)	all (%)
0			7.94	9.56
1			3.64	4.79
2			3.58	4.60
3			3.55	4.56

Fig. 4. The experimental results of the Adirondack image for $n = 0$ until $n = 3$ iterations. The edges are well-preserved for the third iterations and the errors are also decreased.

3. Experimental results

Experiments were carried out to evaluate the performance of the proposed algorithm. Three different datasets have been used. They are the indoor scenes from the Middlebury V3 database [45], the outdoor dataset from the KITTI database [46] and images from Universiti Sains Malaysia Laboratory (USMLab). The quantitative results of experiments in this work will be presented by using

the on-line Middlebury database. This is obtained by uploading the disparity maps into their on-line system. The Middlebury dataset consists of 30 images (i.e., 15 training images and 15 testing images). These training images are developed to determine the parameters in an algorithm and can be uploaded several times. However, the testing images are only for the final evaluation. The KITTI images were taken from the real world of vehicle navigations. The measurements in this work were based on the development kit

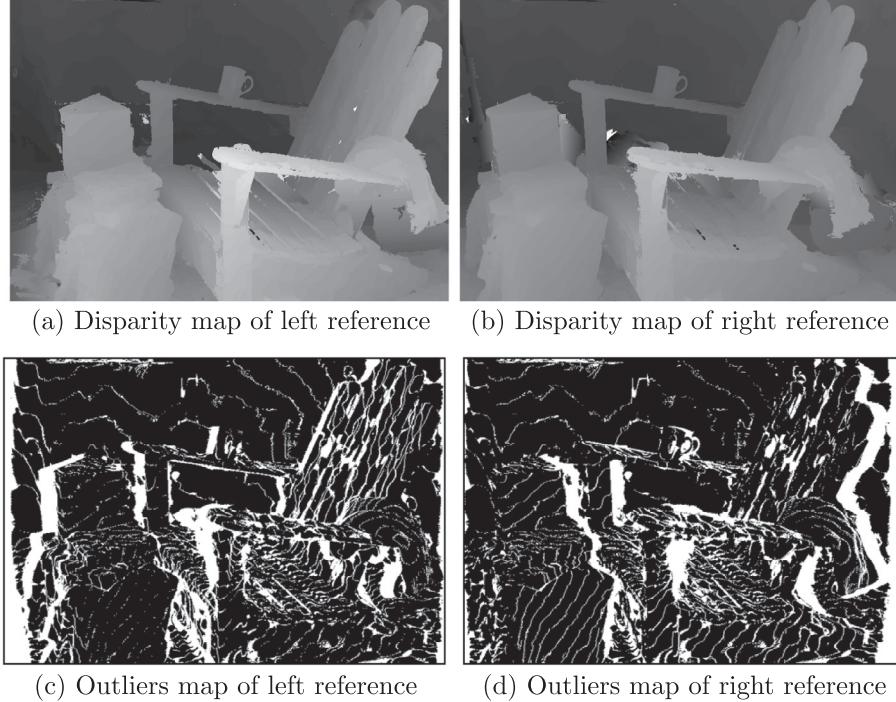
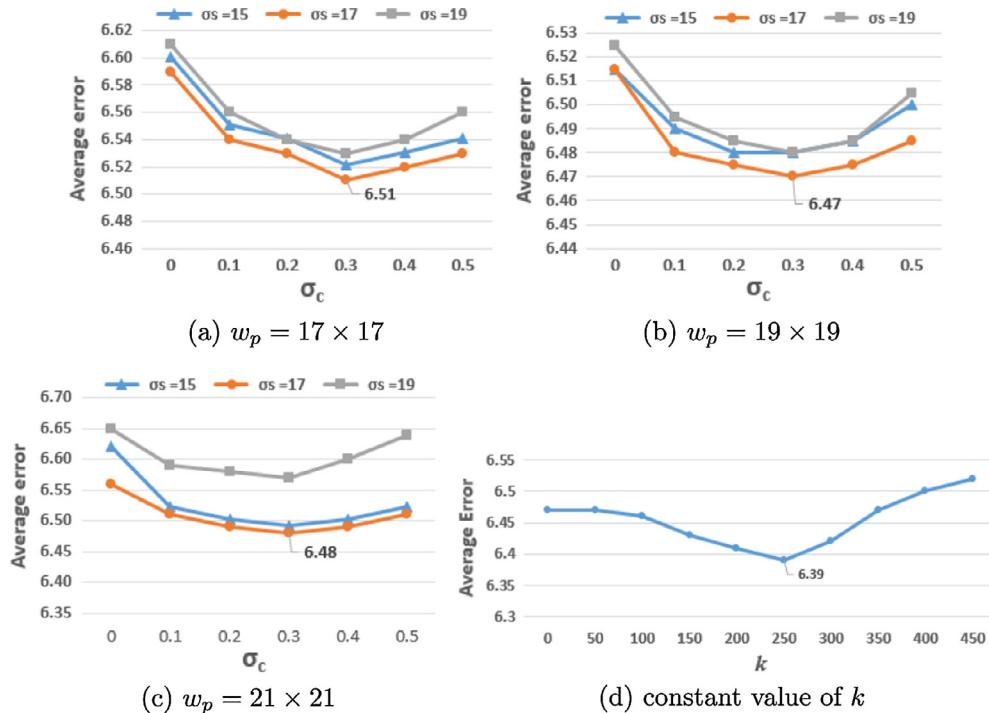


Fig. 5. The experimental results of LR consistency checking process on the Adirondack image. The τ_{LR} value is 0.



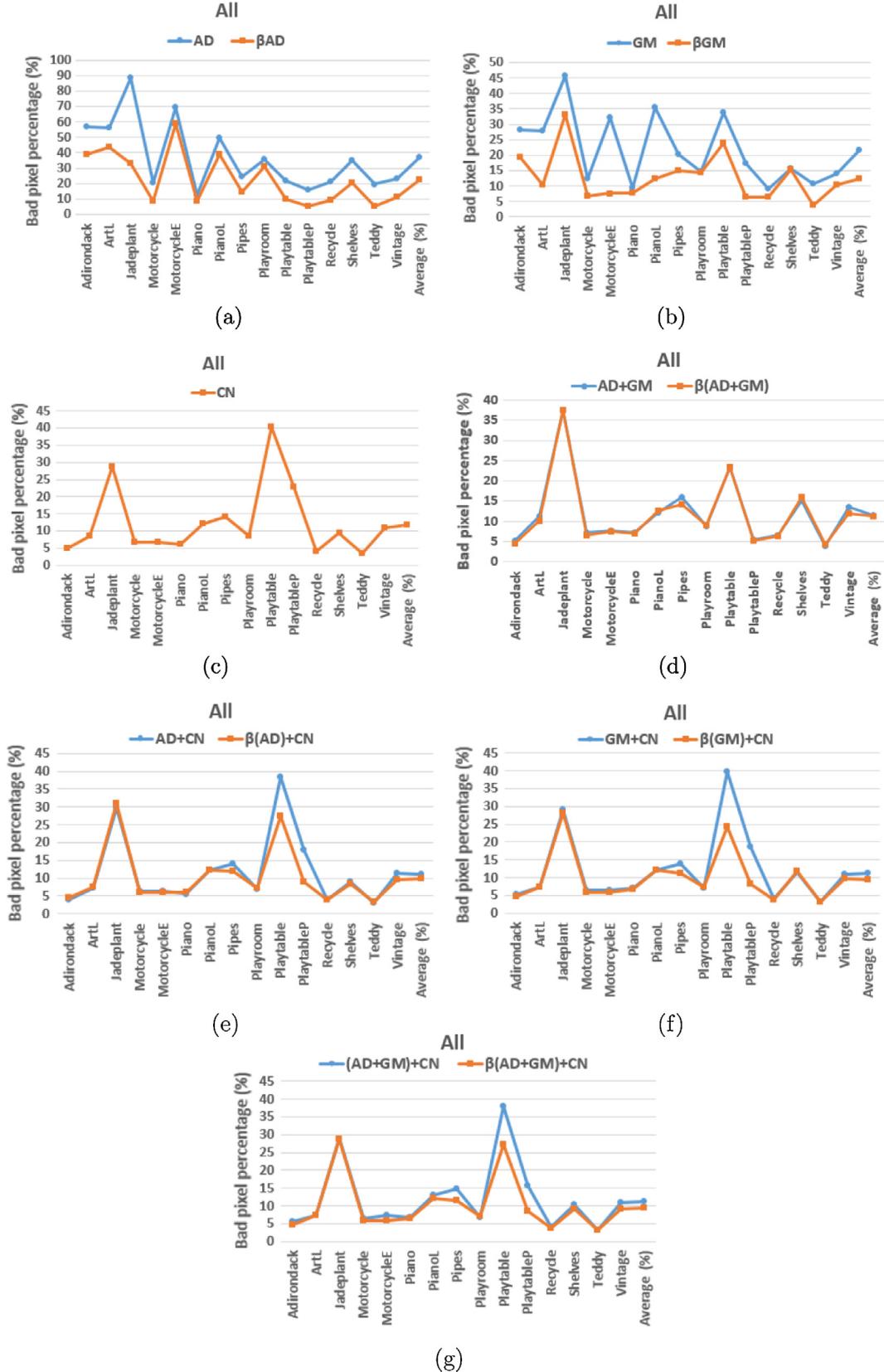


Fig. 7. Performance comparison of the single and combined matching costs using the Middlebury dataset based on error of *all* attribute. The results also consist of the proposed β element in each matching cost. The average of bad pixel percentage for single matching cost (a) AD = 37.30%, β AD = 22.50%; (b) GM = 21.50%, β GM = 12.40%; (c) CN = 12.30% and combined matching costs; (d) AD + GM = 13.90%, β (AD + GM)=11.21%; (e) AD + CN = 11.00%, β (AD)+CN = 9.76%; (f) GM + CN = 11.30%, β (GM)+CN = 9.56%; (g) AD + GM + CN = 11.20%, β (AD + GM)+CN = 9.49%.

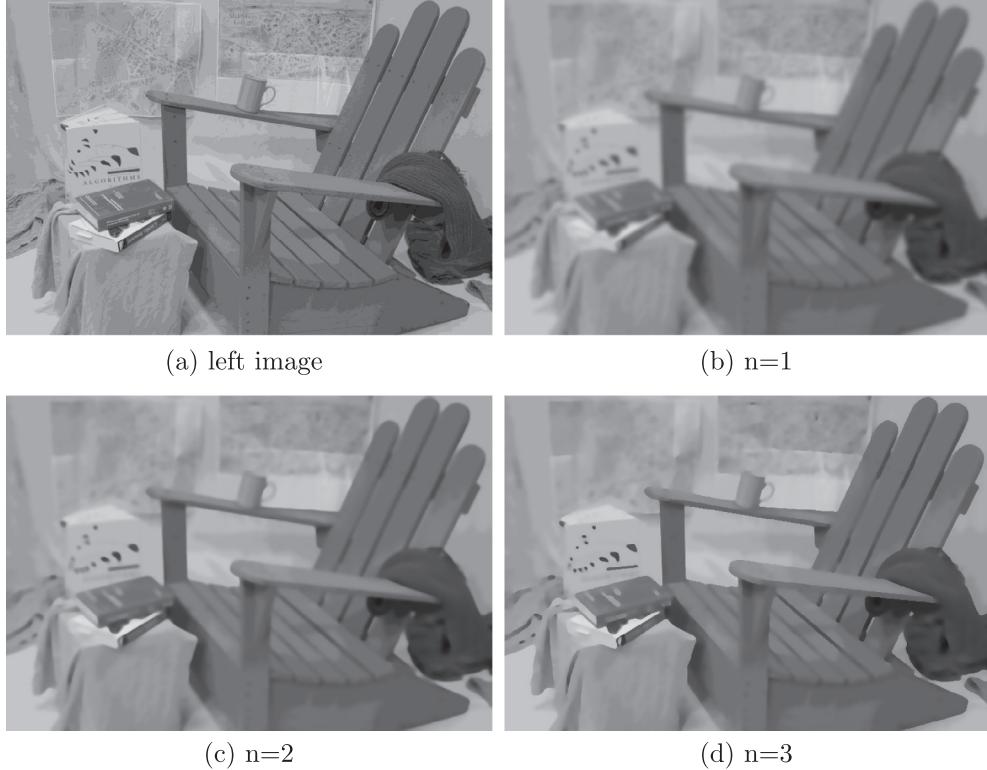


Fig. 8. The iteration results of the guidance grayscale image of Adirondack. The 8d image shows an improved quality compared with images in 8b and c.

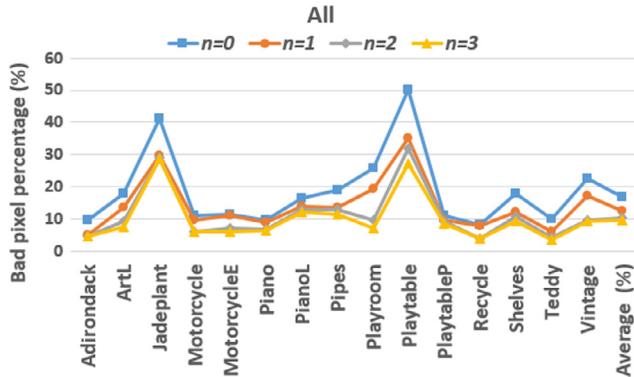


Fig. 9. The results of the iterative GF based on *all* error. The average error of iterations are ($n = 0, n = 1, n = 2, n = 3$) equal to (16.6%, 12.3%, 10.48%, 9.49%).

provided by KITTI Vision Benchmark in [46]. The performance of each disparity map is measured by the percentage of absolute disparity error in two different masks. They are the whole image area *all* pixels and non-occluded area *nonocc*. The images in the USMLab were taken using the Bumblebee stereo camera without any image

enhancement. All of the experiments are running on the central processing unit (CPU) with CORE i5 3.2 GHz and 12 GB of RAM. The proposed algorithm was implemented in C++ programming language and OpenCV library.

3.1. Parameter settings

All the constant parameters in this work are determined from the experiments using the Middlebury training dataset. The measurement is taken from an average error of *nonocc* attribute. Initially, all of the constant parameters are fixed to a minimum value and sequentially started to increase until the output of average error reach to a minimum value. This work has also shown the selection of parameter values for each step of algorithm development. This is to maximize the usage of every parameter. The parameter settings of this work are explained as follows:

Step 1: Fig. 2 shows the results of (a) $\beta = 0.3$ with the lowest average error at 6.96, (parameters set for $\tau_{AD} = 0, \tau_{GM} = 0, \alpha = 0$); (b) $\tau_{AD} = 0.02$ with the lowest average error at 6.89, (parameters set for $\beta = 0.3, \tau_{GM} = 0, \alpha = 0$); (c) $\tau_{GM} = 0.008$ with the lowest average error at 6.87, (param-

Table 1

The comparison results of *all* error for the proposed algorithm and three other different methods on Step 2.

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorcycE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge	Weight ave
Proposed algorithm	4.56	7.33	28.80	5.87	5.91	6.36	12.10	11.50	7.16	27.30	8.64	3.85	9.19	3.30	9.12	9.49
Step 2 (ASW GF) [24]	4.54	7.32	28.50	5.90	5.98	6.36	12.10	11.90	6.76	28.10	8.83	3.91	9.04	3.32	10.50	9.59
Step 2 (non-local) [48]	4.29	7.25	26.70	6.21	6.23	6.20	9.16	12.40	8.79	27.20	9.97	4.33	6.87	3.23	19.20	9.79
Step 2 (ST) [16]	4.59	17.20	27.30	6.26	6.12	6.67	13.2	11.3	7.73	20.10	6.10	3.90	8.99	3.60	9.64	9.84

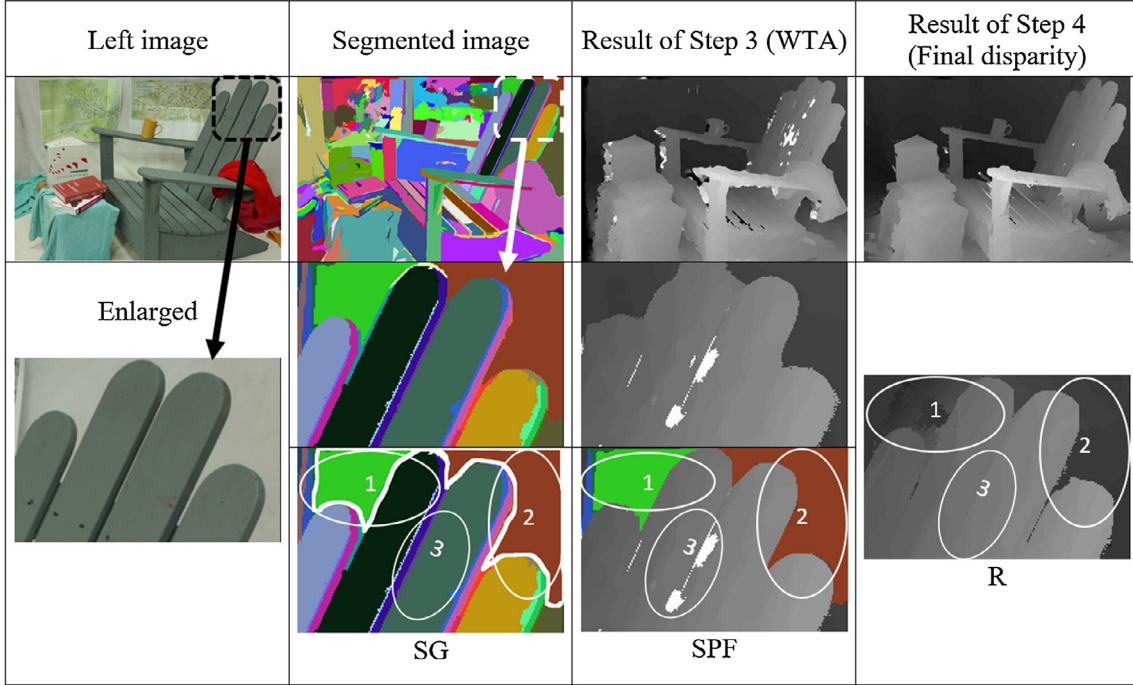


Fig. 10. The left image is enlarged at the top corner of the Adirondack image which is shown by the black arrow. The segmented image is enlarged at the same location which is shown by the white arrow. The region 1, 2 and 3 indicate the uniform regions of segmented area and the area to be uniformed at Step 3. The result of Step 3 is the disparity map without uniform region structure. Result of Step 4 is the final disparity map. The enlarged SG image is the regions with $\tau_{plane} \leq 0.2$ denoted with the region 1, 2 and 3. The SPF image is the region of plane fitting area. The R image is the enlarge disparity map for the regions that meet the $\tau_{plane} \leq 0.2$.

Table 2
The results of *all* error based on with and without the segmentation process. The results are also included with mean shift segmentation technique (MeanShiftSeg) at Step 2 for comparison.

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge	Weight ave
Without segmentation	4.85	8.57	32.10	7.09	7.02	7.21	12.87	13.90	7.53	26.90	8.70	3.88	9.00	4.58	10.7	10.10
With MeanShiftSeg [3]	3.74	7.43	27.40	5.97	5.85	5.73	11.50	11.40	7.90	29.7	10.50	3.80	8.96	3.28	10.60	9.55
With proposed segmentation	4.56	7.33	28.80	5.87	5.91	6.36	12.10	11.50	7.16	27.30	8.64	3.85	9.19	3.30	9.12	9.49

Table 3
The results of the Middlebury dataset based on *all* error for every step of algorithm development.

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge
Step 1 + Step 3	33.6	52.10	202.00	43.25	44.00	45.70	91.20	83.10	53.00	198.23	64.40	27.24	60.70	23.40	67.20
Step 1 + Step 2 + Step 3	9.1	17.00	53.47	10.72	12.79	11.00	23.20	23.00	14.23	48.53	18.90	8.10	17.20	6.78	18.00
Step 1 + Step 2 + Step 3 + Step 4	4.56	7.33	28.80	5.87	5.91	6.36	12.10	11.50	7.16	27.30	8.64	3.85	9.19	3.30	9.12

(Step 1: Matching cost computation; Step 2: Cost aggregation; Step 3: WTA; Step 4: Disparity refinement).

eters set for $\beta = 0.3$, $\tau_{AD} = 0.02$, $\alpha = 0$; (d) $\alpha = 0.18$ with the lowest average error at 6.76 (parameters set for $\beta = 0.3$, $\tau_{AD} = 0.02$, $\tau_{GM} = 0.008$) and (e) $w_{CN} = 7$ with the lowest average error at 6.67. (Step 2 and Step 4 parameters at this stage are $(w_k, \epsilon, n, w_p, \sigma_s, \sigma_c, k, \tau_{plane})$ equal to $((7 \times 7), 0, 0, (17 \times 17), 15, 0, 0, 0.2)$.)

Step 2: Fig. 3 shows the GF parameters at the cost aggregation step. The lowest average error at this stage is 6.49 at $w_k = 9 \times 9$ and $\epsilon = 0.0001$. The effectiveness of the proposed iterative technique at this stage is shown by Fig. 4 which the edges are well-preserved. The figure shows a sample of Adirondack image which is filtered iteratively. In this work, the iteration is selected at $n = 3$ which pro-

duces the lowest error. (Step 1 parameters are using the final values from the above step and Step 4 parameters at this stage are $(w_p, \sigma_s, \sigma_c, k, \tau_{plane})$ equal to $((17 \times 17), 15, 0, 0, 0.2)$).

Step 3: In this work, the disparity selection uses the WTA strategy.

Step 4: Fig. 5 shows the results of LR consistency checking process. It consists of valid and invalid pixels with black and white color respectively. The white color pixels will be replaced by valid pixel values in the process of fill-in invalid pixels as explained in Section 2.4. There are four constant parameters need to be determined at this stage. The weighted bilateral filter radius w_p is simulated at the appropriate range of (17×17) , (19×19) and (21×21) while the σ_s

is started from 15, 17 and 19. The range of σ_c is 0 to 0.5. Fig. 6b shows that the lowest average error is 6.47 which displayed at w_p equals to (19×19) , σ_s and σ_c equal to

17 and 0.3 respectively. For the constant parameter in the segmentation process, the range of k is simulated within the range of 0 to 450 as shown in Fig. 6d. The low-

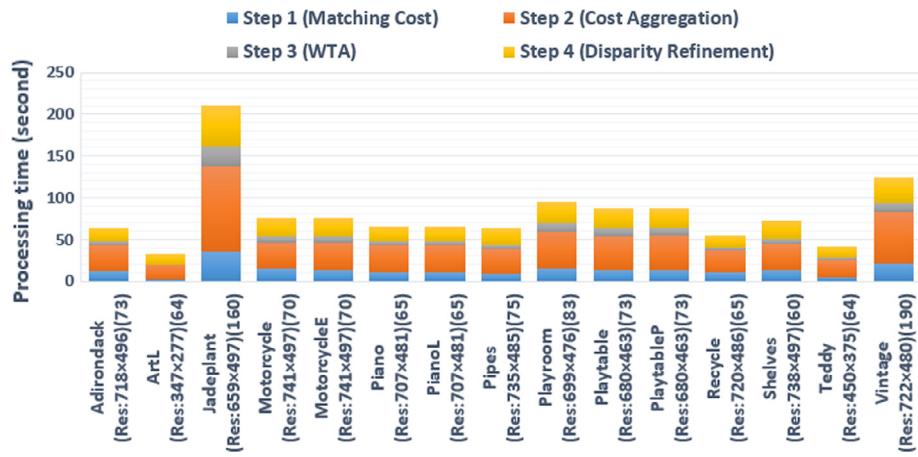


Fig. 11. The execution time of the Middlebury training dataset. Each image is specified with the (Res:resolution) and (maximum disparity range).

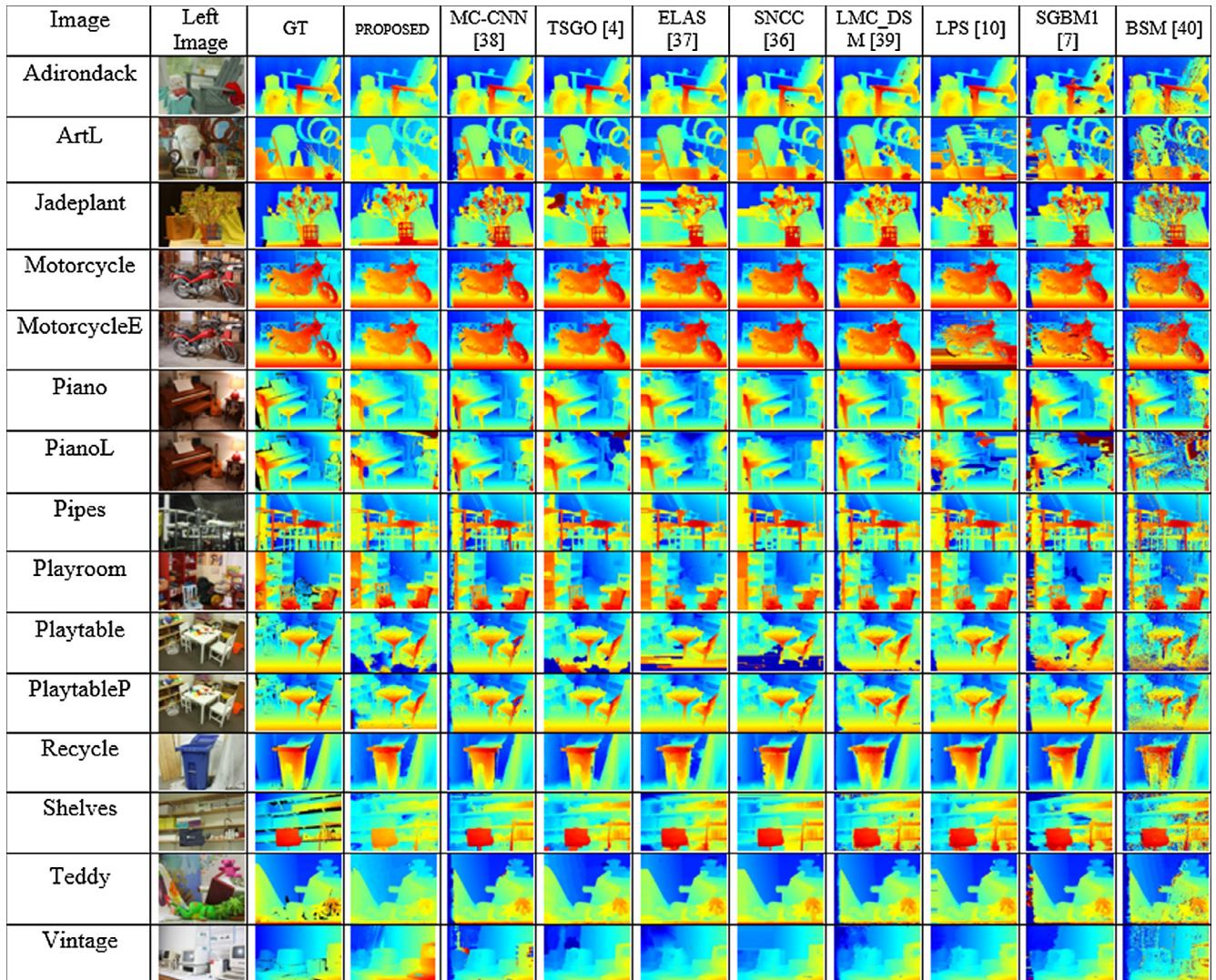


Fig. 12. The disparity maps of the proposed algorithm and other methods on the Middlebury dataset.

Table 4Performance comparison of quantitative evaluation results based on *all* error from the Middlebury dataset.

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge	Weight ave
Proposed algorithm	4.56	7.33	28.80	5.87	5.91	6.36	12.10	11.50	7.16	27.30	8.64	3.85	9.19	3.30	9.12	9.49
TSGO [4]	2.41	4.71	55.6	5.02	4.77	3.50	16.6	8.88	5.69	20.7	2.95	2.66	8.86	2.88	13.5	10.1
SNCC [50]	3.63	6.78	39.8	5.12	5.11	4.65	8.23	11.8	8.05	45.60	4.36	3.29	8.10	2.55	14.80	10.40
ELAS [51]	4.08	7.18	52.8	5.39	5.45	4.96	9.00	10.7	7.94	23.20	3.83	3.78	9.46	3.34	11.60	10.60
MC-CNN [52]	4.24	18.70	34.1	7.21	7.22	6.00	9.35	13.5	18.30	9.71	9.37	4.64	6.62	9.35	21.60	11.80
LPS [10]	5.49	15.50	44.4	4.91	29.70	4.66	21.10	11.9	8.56	8.83	6.15	4.70	10.80	4.16	7.30	12.80
LAMC_DSM [53]	7.65	21.80	37.9	11.30	11.10	8.81	11.70	17.4	22.70	15.40	10.60	5.91	13.40	10.20	15.50	14.60
SGBM1 [7]	8.16	17.50	43.4	11.40	15.40	12.03	29.20	15.1	23.30	13.10	11.40	5.54	14.10	12.20	18.20	16.10
BSM [54]	12.70	28.70	58.7	14.80	14.70	16.00	35.80	24.5	29.40	31.00	20.20	12.10	19.20	14.30	39.30	23.50

Table 5Performance comparison of quantitative evaluation results based on *nonocc* error from the Middlebury dataset.

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge	Weight ave
MC-CNN [52]	0.76	2.55	16.3	1.27	1.27	1.83	5.07	2.29	2.27	3.11	3.03	2.48	4.41	1.12	14.8	3.82
LAMC_DSM [53]	3.54	5.10	18.2	3.26	3.18	4.87	8.74	6.87	7.08	9.38	4.65	2.73	11.7	2.68	10.5	6.31
Proposed algorithm	3.55	4.75	12.40	3.27	3.27	5.41	11.30	6.13	5.03	25.80	5.60	3.36	8.93	2.54	8.11	6.39
SNCC [50]	2.89	4.05	18.1	2.68	2.52	3.52	7.08	6.14	5.64	45.4	3.13	2.90	7.59	1.58	13.5	6.97
TSGO [4]	2.02	3.07	32.5	3.12	2.94	2.96	16.1	4.90	4.02	18.7	2.20	2.33	8.34	2.46	12.6	7.07
ELAS [51]	3.09	4.72	29.7	3.28	3.29	4.30	8.31	5.61	6.00	21.8	2.84	3.09	9.00	2.36	10.9	7.22
LPS [10]	2.15	12.6	15.4	1.45	27.6	3.40	20.1	3.63	4.60	3.24	3.15	2.44	9.51	1.83	4.76	7.58
SGBM1 [7]	4.74	7.54	22.0	3.68	7.28	6.67	26.00	6.76	7.00	6.38	4.62	2.85	11.3	4.63	10.7	8.12
BSM [54]	7.27	11.4	30.5	6.67	6.52	10.8	32.1	10.5	12.5	24.4	12.8	7.42	16.4	4.88	32.8	13.4

est average value is 6.39 at k equals to 250. The τ_{plane} is 0.2 as implemented in [44]. (Step 1 and 2 are using the final values from the above steps.)

3.2. Evaluation based on every step development

Every step of algorithm development requires an evaluation to show the achievement. All related parameters were fixed to the same values which were determined in Section 3.1. Step 1: The evaluations are based on combined matching costs and the efficiency of β element in the framework. Fig. 7 shows the comparative results of a single and combined matching cost functions. The algorithms consist of absolute differences (AD), gradient differences (GM), census transform (CN) and combined matching costs (i.e., AD + GM, AD + CN [47], GM + CN, AD + GM + CN). The lowest average error is produced by the proposed algorithm in Fig. 7(g) with 9.49% which comprises of three matching costs. It shows the effectiveness of combined matching costs which is able to increase the accuracy. Moreover, the proposed algorithm with β element is able to reduce the errors based on the results of with and without β element in each evaluated algorithm. This demonstrates the efficiency of β element in the framework.

Fig. 8 shows the results of the guidance grayscale image at $n = 1$, 2, 3. The $n = 3$ displays a better quality image compared to $n = 1$ and 2. Fig. 9 shows the results of iterative GF at Step 2. The 3rd-iteration is the lowest average error with 9.49%. Table 1 shows the performance comparison of other approaches at Step 2 (i.e., cost aggregation). The framework of Step 1, Step 3 and Step 4 used the same settings which was implemented in the proposed work. The approaches used the same parameter values in [48] (i.e., NL), [16] (i.e., ST) and [24] (i.e., ASW GF). The iterative GF produces the lowest average error. Fig. 10 shows an example of segmented and plane fitting regions of the Adirondack image. Result of Step 3 is disparity map without applying the uniform areas handling. Step 4 is final disparity map which is satisfied the $\tau_{plane} \leq 0.2$ and

consider the planar fitting as successful. Table 2 shows the results of with and without the segmentation process at Step 4. The efficiency of undirected graph segmentation was compared with the mean shift segmentation (MeanShiftSeg) which was developed by Comaniciu and Meer [49]. The rest of the steps were using the same settings unless the segmentation part which was replaced by the MeanShiftSeg technique with the usage parameters (i.e., $(h_s, h_r, M) = (16, 7, 40)$). These parameters were used based on the experimental results in this work which produced at the lowest average on both errors (i.e., *all* and *nonocc* errors). The average running time of 15 images in Table 2 are 65 s for without the segmentation, 75 s for the proposed work and 81 s for the MeanShiftSeg method.

This work also presented the error reductions and execution time in every stage of algorithm development. Table 3 shows the results of *all* error based on the sequential steps of the algorithm development. Basically, Step 1 produces high error. These errors reduced tremendously once the Step 2 (i.e., filtering stage) takes place. Then, Step 4 refines and removes the remaining noise which produces the final disparity maps. Fig. 11 shows the results of the computational time for the Middlebury training dataset. In average, the longest time execution in the proposed algorithm is at Step 2.

3.3. Middlebury dataset

The final results (i.e., using the Middlebury dataset) are displayed in Fig. 12 with all training images, ground truth (GT) and the proposed algorithm. The performance of the proposed algorithm is also compared with other established stereo matching algorithms in [32] through Fig. 12, Tables 4 and 5. These tables summarize the quantitative performance in descending order of overall performance. These algorithms are compared due to their different approaches in the development. Table 4 shows the results of *all* error and the proposed method is highly ranked among the

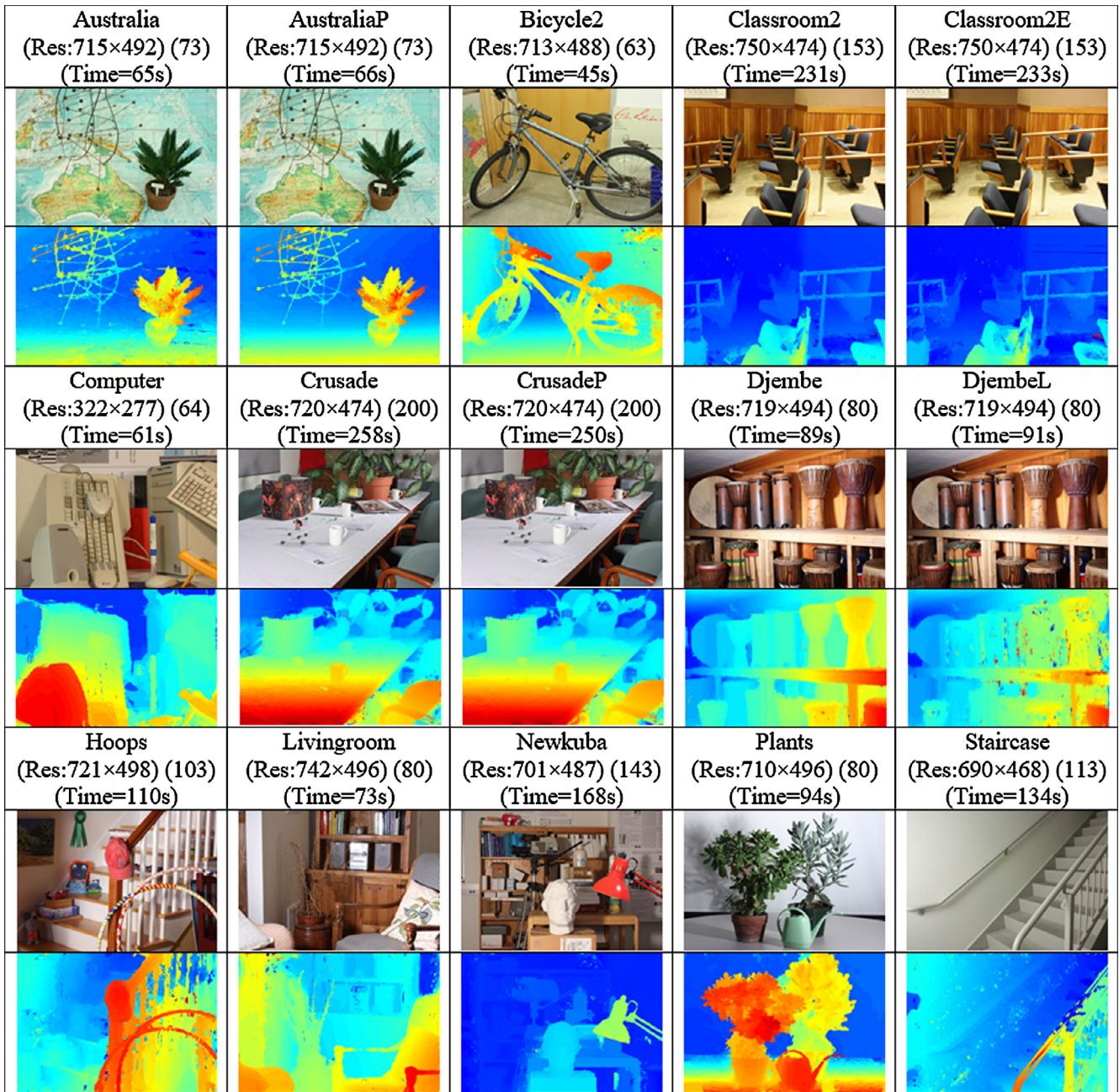


Fig. 13. The disparity maps of the Middlebury testing dataset. Each image displays the resolution (Res), (maximum disparity) and time execution (Time).

local and semi-global methods. Table 5 shows the results of *nonocc* error which the proposed algorithm is ranked in the top three. It can be seen that the results of the proposed algorithm for the Adirondack, Recycle and Vintage images, the foreground objects are detached from the background with clear and precise contours and accurate disparity values in accordance with their depth order. For the complex scene objects (i.e., Jadeplant, Motorcycle, Playroom, Piano and Shelves stereo pairs), the disparity values of the

layered objects are correctly reconstructed in accordance with their respective depths. The scene objects situated at increasing depth are assigned step by step to disparity values from nearer to further. The competitive achievement of the proposed algorithm shows that the aim of this work to develop an accurate local stereo matching algorithm has succeeded. Fig. 13 shows the results of the Middlebury testing dataset. The smooth disparity maps were produced for the images of Australia, AustraliaP, Bicycle2, Classroom2,

Table 6

The results of the Middlebury dataset based on the percentage of disparity errors that are greater than two pixels (bad 2.0).

Algorithms	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlayP	Recyc	Shelvs	Teddy	Vintge
<i>all</i> error															
Proposed algorithm	39.30	36.20	53.50	35.60	37.90	48.60	55.40	42.50	50.60	62.20	48.40	40.20	53.60	26.90	57.60
<i>nonocc</i> error															
Proposed algorithm	36.70	28.00	44.60	31.50	33.70	45.30	52.50	33.50	44.40	59.80	44.40	37.80	52.50	21.60	54.90

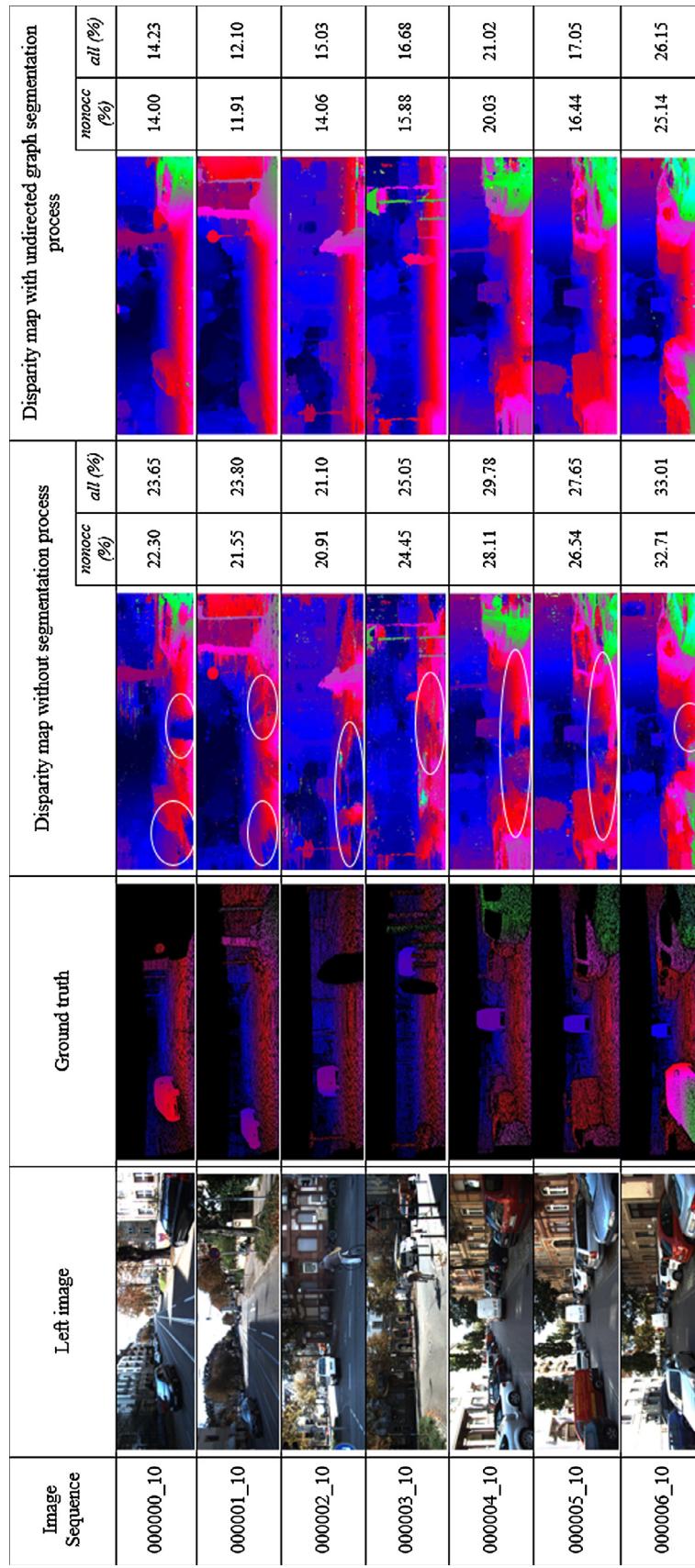


Fig. 14. The results of the KITTI dataset. These sample training images are numbered sequentially according to the database. The proposed algorithm is able to reduce both errors (i.e., *nonocc* and *all*).



Fig. 15. The results of the USMLab images.

Classroom2E, Computer, Crusade, CrusadeP, Djembe, Livingroom, Newkuba and Plants. **Table 6** shows the percentage results of bad pixels with disparity errors greater than two pixels (bad 2.0) on the *all* and *nonocc* errors. This evaluation metric is based on the threshold value set by the Middlebury database.

3.4. KITTI

The experiments on other datasets to test for more reliable evaluation of the proposed algorithm are also conducted. It is to show the adaptability of the proposed algorithm with more challenging of real stereo images. All images from KITTI were acquired from the real world conditions. Therefore the stereo images contain large textureless regions (e.g., walls, sky, road and inconsistent brightness conditions e.g., light of the sun, shades of tree and other moving objects). All the images contains a similar size of resolution (i.e., Res: 1242×375) and maximum disparity value (i.e., 256). **Fig. 14** shows the left reference images, ground truth and the results of the proposed algorithm for the image sequence 000000_10-000006_10 of the KITTI dataset. The time execution of a single image is 220 s. The proposed algorithm produced smooth disparity maps and contain details of edge contour according to the objects detected. The low textured regions are well recovered compared to the algorithm without the segmentation process as shown by the white circles in **Fig. 14**.

3.5. USMLab

Fig. 15 shows the results of the USMLab images. The results demonstrate smooth disparity maps with clear and detailed edge

contours. The resolution of the images are (512×384), maximum disparity value (64) and the execution time of an image is 58 s.

4. Conclusion

In this work, the new local-based stereo matching algorithm is presented. The proposed algorithm is able to reduce the errors and increase the accuracy based on the Middlebury and KITTI datasets. The combination of matching cost functions based on (AD + GM + CN) is able to overcome the disadvantage of a single matching cost. Furthermore, the proposed iterative GF is able to improve and preserve the edges. The usage of undirected graph segmentation with plane-fitting process increases the robustness against the low texture areas. The results of disparity maps have demonstrated that the proposed method not only perform well on the standard indoor dataset, but also on the stereo images of the outdoor scenes. In future, the proposed work will be implemented on GPU to reduce the execution time.

Acknowledgement

This work was supported by Universiti Sains Malaysia (No: PLD-0025/13(R)) and Universiti Teknikal Malaysia Melaka.

References

- [1] M. Humenberger, T. Engelke, W. Kubinger, A census-based stereo vision algorithm using modified semi-global matching and plane fitting to improve matching quality, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 77–84.

- [2] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Comput. Vision* 47 (1) (2002) 7–42.
- [3] Y. Zhou, C. Hou, Stereo matching based on guided filter and segmentation, *Optik* 126 (9) (2015) 1052–1056.
- [4] M.G. Mozerov, J. van de Weijer, Accurate stereo matching by two-step energy minimization, *IEEE T. Image Process.* 24 (3) (2015) 1153–1163.
- [5] H.Q. Wang, M. Wu, Y.B. Zhang, L. Zhang, Effective stereo matching using reliable points based graph cut, in: *Proceedings of Visual Communications and Image Processing*, 2013, pp. 1–6.
- [6] X. Wang, H. Wang, Y. Su, Accurate belief propagation with parametric and non-parametric measure for stereo matching, *Optik* 126 (5) (2015) 545–550.
- [7] H. Hirschmüller, Stereo processing by semiglobal matching and mutual information, *IEEE T. Pattern Anal.* 30 (2) (2008) 328–341.
- [8] K. Wenzel, M. Rothermel, D. Fritsch, N. Haala, Image acquisition and model selection for multi-view stereo, *Int. Arch. Photogram. Rem. Sens. Spatial. Inform. Sci.* XL-5 (2013) 251–258.
- [9] F. Bethmann, T. Luhmann, Semi-global matching in object space, *Int. Arch. Photogram. Rem. Sens. Spatial. Inform. Sci.* 40 (3) (2015) 23–30.
- [10] S.N. Sinha, D. Scharstein, R. Szeliski, Efficient high-resolution stereo matching using local plane sweeps, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1582–1589.
- [11] G.W. Zheng, X.H. Jiang, A fast stereo matching algorithm based on fixed-window, *Appl. Mech. Mater.* 411 (2013) 1305–1313.
- [12] H. Hirschmüller, P.R. Innocent, J. Garibaldi, Real-time correlation-based stereo vision with reduced border errors, *Int. J. Comput. Vision* 47 (1–3) (2002) 229–246.
- [13] S. Zhu, Z. Li, Local stereo matching using combined matching cost and adaptive cost aggregation, *KSII T. Internet Inf.* 9 (1) (2015) 224–241.
- [14] G.A. Kordelas, D.S. Alexiadis, P. Daras, E. Izquierdo, Enhanced disparity estimation in stereo images, *Image Vision Comput.* 35 (2015) 31–49.
- [15] C. Yang, Y. Li, W. Zhong, S. Chen, Real-time hardware stereo matching using guided image filter, in: *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*, ACM, 2016, pp. 105–108.
- [16] X. Mei, X. Sun, W. Dong, H. Wang, X. Zhang, Segment-tree based cost aggregation for stereo matching, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 313–320.
- [17] X. Tan, C. Sun, X. Sirault, R. Furbank, T.D. Pham, Stereo matching using cost volume watershed and region merging, *Signal Process.: Image Commun.* 29 (10) (2014) 1232–1244.
- [18] K.-J. Yoon, I.S. Kweon, Adaptive support-weight approach for correspondence search, *IEEE T. Pattern Anal.* 28 (4) (2006) 650–656.
- [19] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, C. Rother, Real-time local stereo matching using guided image filtering, in: *International Conference on Multimedia and Expo (ICME)*, 2011, pp. 1–6.
- [20] K. He, J. Sun, X. Tang, Guided image filtering, in: *European Conference on Computer Vision*, Springer, 2010, pp. 1–14.
- [21] K. He, J. Sun, X. Tang, Guided image filtering, *IEEE Trans. Pattern Anal. Machine Intell.* 35 (6) (2013) 1397–1409.
- [22] J. Zhang, J.-F. Nezan, M. Pelcat, J.-G. Cousin, Real-time gpu-based local stereo matching method, in: *Design and Architectures for Signal and Image Processing (DASIP)*, 2013, pp. 209–214.
- [23] Q. Yang, P. Ji, D. Li, S. Yao, M. Zhang, Fast stereo matching using adaptive guided filtering, *Image Vision Comput.* 32 (3) (2014) 202–211.
- [24] S. Zhu, R. Gao, Z. Li, Stereo matching algorithm with guided filter and modified dynamic programming, *Multimed. Tools Appl.* (2015) 1–18.
- [25] S. Zhu, Z. Wang, X. Zhang, Y. Li, Edge-preserving guided filtering based cost aggregation for stereo matching, *J. Visual Commun. Image Represent.* 39 (2016) 107–119.
- [26] G.A. Kordelas, D.S. Alexiadis, P. Daras, E. Izquierdo, Content-based guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation, *IEEE Trans. Multimedia* 18 (2) (2016) 155–170.
- [27] R.A. Hamzah, H. Ibrahim, Literature survey on stereo vision disparity map algorithms, *J. Sensors* (2015) 1–23.
- [28] H. Jin, S. Liu, S. Zhang, G. Ying, Superpixel-based global optimization method for stereo disparity estimation, in: *Pattern Recognition*, 2014, pp. 445–454.
- [29] D. Chen, M. Ardabiliyan, L. Chen, A fast trilateral filter-based adaptive support weight method for stereo matching, *IEEE T. Circ. Syst. Vid.* 25 (5) (2015) 730–743.
- [30] Y. Zhan, Y. Gu, K. Huang, C. Zhang, K. Hu, Accurate image-guided stereo matching with efficient matching cost and disparity refinement, *IEEE T. Circ. Syst. Vid.* (2015) 1–14.
- [31] Y. Peng, G. Li, R. Wang, W. Wang, Stereo matching with space-constrained cost aggregation and segmentation-based disparity refinement, in: *Three-Dimensional Image Processing, Measurement, and Applications*, 2015, pp. 939309–939315.
- [32] D. Scharstein, R. Szeliski, Middlebury stereo evaluation – version 3, <<http://vision.middlebury.edu/stereo/eval/references>> (accessed date: March 2016).
- [33] S. Mattoccia, F. Tombari, L. Di Stefano, Stereo vision enabling precise border localization within a scanline optimization framework, in: *Asian Conference on Computer Vision*, Springer, 2007, pp. 517–527.
- [34] P. Tan, P. Monasse, Stereo disparity through cost aggregation with guided filter, *Image Process. On Line (IPOL)* 4 (2014) 252–275.
- [35] S. Ploumpis, A. Amanatidis, A. Gasteratos, A stereo matching approach based on particle filters and scattered control landmarks, *Image Vision Comput.* 38 (2015) 13–23.
- [36] R.G. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, Lsd: a line segment detector, *Image Process. On Line (IPOL)* 2 (3) (2012) 35–55.
- [37] Z. Lee, J.-C. Juang, T.Q. Nguyen, Local disparity estimation with three-moded cross census and advanced support weight, *IEEE T. Multimedia* 15 (8) (2013) 1855–1864.
- [38] K. Zhang, J. Lu, Q. Yang, G. Lafruit, R. Lauwereins, L. Van Gool, Real-time and accurate stereo: a scalable approach with bitwise fast voting on cuda, *IEEE T. Circ. Syst. Vid.* 21 (7) (2011) 867–878.
- [39] S. Lee, J.H. Lee, J. Lim, I.H. Suh, Robust stereo matching using adaptive random walk with restart algorithm, *Image Vision Comput.* 37 (2015) 1–11.
- [40] P.F. Felzenswalb, D.P. Huttenlocher, Efficient graph-based image segmentation, *Int. J. Comput. Vision* 59 (2) (2004) 167–181.
- [41] Y. Wang, F. Zhong, Q. Peng, X. Qin, Depth map enhancement based on color and depth consistency, *Visual Comput.* 30 (10) (2014) 1157–1168.
- [42] D. Min, J. Lu, M.N. Do, Depth video enhancement based on weighted mode filtering, *Pattern Recogn. Lett.* 21 (3) (2012) 1176–1190.
- [43] Z. Ma, K. He, Y. Wei, J. Sun, E. Wu, Constant time weighted median filtering for stereo matching and beyond, in: *IEEE International Conference on Computer Vision*, 2013, pp. 49–56.
- [44] M. Humenberger, T. Engelke, W. Kubinger, A census-based stereo vision algorithm using modified semi-global matching and plane fitting to improve matching quality, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2010, pp. 77–84.
- [45] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, P. Westling, High-resolution stereo datasets with subpixel-accurate ground truth, in: *Pattern Recognition*, 2014, pp. 31–42.
- [46] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.
- [47] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, X. Zhang, On building an accurate stereo matching system on graphics hardware, in: *IEEE International Conference on Computer Vision Workshops*, IEEE, 2011, pp. 467–474.
- [48] Q. Yang, A non-local cost aggregation method for stereo matching, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2012, pp. 1402–1409.
- [49] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, *IEEE Trans. Pattern Anal. Machine Intell.* 24 (5) (2002) 603–619.
- [50] N. Einecke, J. Eggert, Anisotropic median filtering for stereo disparity map refinement, in: *International Conference on Computer Vision Theory and Applications*, 2013, pp. 189–198.
- [51] A. Geiger, M. Roser, R. Urtasun, Efficient large-scale stereo matching, in: *Computer Vision (ACCV)*, 2011, pp. 25–38.
- [52] J. Zbontar, Y. LeCun, Computing the stereo matching cost with a convolutional neural network, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1592–1599.
- [53] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, G. Karras, On accurate dense stereo-matching using a local adaptive multi-cost approach, *ISPRS J. Photogramm.* 91 (2014) 29–49.
- [54] K. Zhang, J. Li, Y. Li, W. Hu, L. Sun, S. Yang, Binary stereo matching, in: *International Conference on Pattern Recognition (ICPR)*, 2012, pp. 356–359.