EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Shape-reserved stereo matching with segment-based cost aggregation and dual-path refinement

Chih-Shuan Huang[1], Ya-Han Huang[1], Din-Yuen Chan[2] and Jar-Ferr Yang[1*]

* Correspondence: jefyang@mail.ncku.edu.tw
[1]Department of Electrical Engineering, National Cheng Kung University, 1, University Rd, Tainan City, Taiwan
Full list of author information is available at the end of the article

Abstract

Stereo matching is one of the most important topics in computer vision and aims at generating precise depth maps for various applications. The major challenge of stereo matching is to suppress inevitable errors occurring in smooth, occluded, and discontinuous regions. In this paper, the proposed stereo matching system uses segment-based superpixels and matching cost. After determination of edge and smooth regions and selection of matching cost, we suggest the segment-based adaptive support weights in cost aggregation instead of color similarity and spatial proximity only. The proposed dual-path depth refinements use the cross-based support region by referring texture features to correct the inaccurate disparities with iterative procedures to improve the depth maps for shape reserving. Specially for leftmost and rightmost regions, the segment-based refinement can greatly improve the mismatched disparity holes. The experimental results show that the proposed system can achieve higher accurate depth maps than the conventional methods.

1 Introduction

With fast evolutions of nature three-dimension (3D) technologies, the applications of mixed reality [1], visual entertainment [2–4], environment reconstruction [5], autonomous driving [6], object detection [7, 8], and recognition [9] with additional depth information become more and more important nowadays. All the above applications, the key part is to retrieve high accuracy depth maps from multiple camera images. Instead of transmitting complex multiple views, a color texture image with its corresponding gray depth map can effectively represent the 3D information. For satisfying 3D vision, the traditional way is to directly provide multi-view/stereo-view videos, but the 2D image plus depth map is a preferable way to characterize 3D sensation nowadays. The depth map provides the pixel-wise distance and exhibits stereoscopic vision. We can use the depth image-based rendering (DIBR) system [10] to create multi-view videos with depth information and texture image in the user side.

The concept of stereo vision comes from the different views at distinct positions of the scene, leads a limited displacement in a pair of corresponding pixels, i.e., so-called

"disparity". The disparity becomes larger when the object is moving toward the observer [11]. By parsing the disparities of left and right views, we can also extend the geometrical principle to estimate the distance between a viewed object and the observer. To get the depth map efficiently, we propose a local stereo matching method to save the computation. Since the depth values are mostly dependent on bases of the objects. By using segmentation information, the proposed stereo matching system can not only enhance the aggregation efficiency but also refine the missing objects. The basic idea of stereo matching will be briefly reviewed in Section 2. In Section 3, the designs of the proposed stereo matching system are present. Section 4 will show the experimental results achieved by the proposed and other methods. Some conclusions about this paper are finally given in Section 5.

2 Local stereo matching methods

Generally, the stereo matching algorithms can be classified into three major categories: global, local, and semi-global approaches. The global approach uses data term and smooth team to construct their energy functions to compute the global depth map. Graph-cut [12], belief propagation [13, 14], and dynamic programming [15] are the typical global stereo matching algorithms. Recently, deep learning approaches have been proposed to estimate the depth maps [16]; however, they are data dependent. In this paper, we focus on the designs of local stereo matching approaches for computation considerations and avoid the problems of data dependency.

2.1 Local stereo matching process

The typical local stereo matching process shown in Fig. 1 mainly contains matching cost computation, cost aggregation, disparity decision, and disparity refinement stages.

2.1.1 Matching cost computation

To evaluate the pixel-based matching status, there are several famous costs that are used for disparity estimation. The sum of absolute differences (SAD) [17] with color components is the most common cost of stereo matching. The SAD cost for finding the left disparity map can be formulated as:

$$C_{SAD}(p,p') = \sum_{c\in\{R,G,B\}} \left|I_c^l(p) - I_c^r(p')\right|, \tag{1}$$

where $p = (x, y)$ is the pixel position in the left image and $p' = (x\text{-}d, y)$ denotes its corresponding pixel position in the right image with disparity d. In this paper, I_c^l and I_c^r represent the color intensities of the left and right images in RGB domain, respectively.

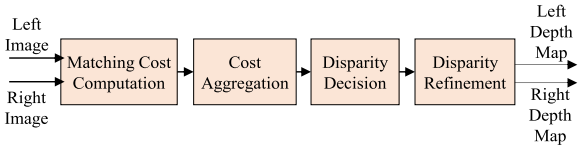

Fig. 1 Flow diagram of typical local stereo matching method

Besides the SAD cost, the gradient similarity can also measure the variations of the texture images. The gradient cost for searching the left disparity map can be expressed by

$$C_{grad}\left(p,p^{'}\right)=\sum_{c\in\{R,G,B\}}\left|\nabla I_c^l(p)-\nabla I_c^r(p')\right|, \tag{2}$$

where the gradient operator, ∇, is the combination of horizontal and vertical differences between the central pixel and its neighboring pixels in the cross relation.

Besides, the census transform, which detects the slight variations in a small block, can achieve a robust performance for minor intensity changes successfully. Figure 2 shows the traditional census and modified mini-census transforms [18]. The modified mini-census transform only selects a few specified representative pixels in the block to reduce the useless information.

To describe the above census transforms precisely, the binary result is expressed by comparing the neighboring pixel to the central pixel at p as

$$\xi(p,q)=\begin{cases}1, & I(p)\le I(q)\\ 0, & I(p)>I(q)\end{cases}, \tag{3}$$

where q denotes the position of the neighboring pixel in the block. A bit-wise catenation is applied to get the census transform as

$$c(p)=\underset{q\in W}{\otimes}\xi(p,q), \tag{4}$$

where $\otimes$ expresses the bit-wise catenation operator and W is the block containing the selected neighboring pixels. Thus, the census cost in terms of Hamming distance between two census transforms obtained from left and right views is expressed by:

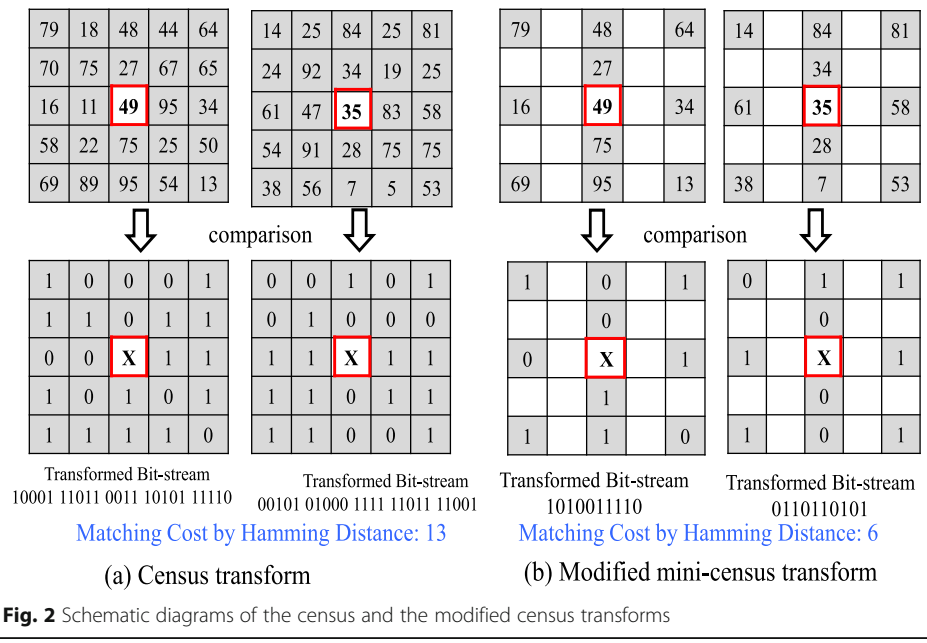

Fig. 2 Schematic diagrams of the census and the modified census transforms

$$C_{\text{census}}\left(p, p^{'}\right) = c_l(p) \oplus c_r\left(p^{'}\right), \tag{5}$$

where $\oplus$ is the bit-wise XOR operation. The modified mini-census transform [18] needs fewer computations and achieves more robust performance against the noises than the traditional census transform.

2.1.2 Cost aggregation

Once the cost of the paired pixels in the stereo images is calculated, the cost aggregation is further applied to achieve more robust results by including more pixels, which have the same tendency. For local stereo matching, the window-based aggregation considers the similarities of the surrounding pixels in a designated window [19–25]. The ideal windows are designed to include the nearby pixels, which are in the same object as possible. For example, the adaptive support weights [21] based on color similarity and spatial proximity are noted as

$$w(p,q) = \exp\left(- \left(\Delta c_{pq}/\gamma_c + \Delta g_{pq}/\gamma_g \right) \right), \tag{6}$$

where Δc_{pq} and Δg_{pq} denote the color similarity weight and the geometric distance weight, respectively. γ_c and γ_g are control factors that map Δc_{pq} and Δg_{pq} to become weights. The color similarity weight is controlled by Δc_{pq}, which can be represented as

$$\Delta c_{pq} = \sum_{c \in \{R,G,B\}} | I_c(p) - I_c(q) |, \tag{7}$$

while the geometric distance weight is controlled by Δg_{pq}, which is given as

$$\Delta g_{pq} = \left(x_p - x_q\right)^2 + \left(y_p - y_q\right)^2, \tag{8}$$

where (x_p, y_p) and (x_q, y_q) are the x and y coordinates of pixels p and q, respectively.

Besides the pixel-wise adaptive support weights, the segmentation concept is also used to modify the weights increasing the matching reliability. The segment-based adaptive support weight [22, 23] could be expressed by

$$w(p,q) = \begin{cases} 1.0 & , \quad \text{if } q \in S_p \\ \exp\left(- \Delta c_{pq}/\gamma_c \right) & , \quad \text{if } q \notin S_p \end{cases}, \tag{9}$$

where S_p is the segment on which p lies. In (9), they modify the weight to the largest, i.e., 1.0 if the neighboring pixel is in the same segment as the target pixel while the weights of the rest pixels are determined by color similarity.

After the weight of each pixel in the window has been calculated, we can apply the aggregation cost to all the pixel costs become as

$$C_{agg}(p,p') = \frac{\sum_{q \in W_r, q' \in W_t} w_r(p,q) \cdot w_t(p',q') \cdot C(p,p')}{\sum_{q \in W_r, q' \in W_t} w_r(p,q) \cdot w_t(p',q')}, \tag{10}$$

where $C(p, p')$ is the initial cost, which could be SAD cost, gradient cost, or census cost stated in (2), (3), or (5), respectively. Of course, the combined cost with different weighting ratios is also possible. In (10), q and q' are the neighboring pixels of p and p'

pixels in the target and the reference windows of the target and the reference views, respectively.

2.1.3 Raw disparity estimation

To obtain the raw disparity map, the disparity estimation is executed after cost aggregation. It is common to utilize the winner-take-all (WTA) strategy for the criterion of disparity estimation. The selection of WTA can be formulated as

$$d_p = \arg \min_{d \in R_d} C_{agg}(p, p'), \tag{11}$$

where R_d is the disparity search range. In the WTA process, we can finally estimate the raw disparity by choosing the smallest cost. The raw disparity map d_p needs to be refined in the final disparity refinement process.

2.1.4 Disparity refinement

Usually, the raw disparity map contains mismatched disparities occurring near the object boundaries due to occlusion problems and the regions with smooth texture regions, which are hard to find the exact matches. Thus, a suitable disparity refinement technique is required to remove the mismatched disparities. First, we need to identify the mismatched pixels by left right consistency check (LRC) to test if the disparities of the left and right views are consistent.

The LRC detection rule is normally stated as

$$L(x,y) = \begin{cases} 1, & |d_l(x,y) - d_r(x - d_L(x,y), y)| < \sigma_0 \\ 0, & |d_l(x,y) - d_r(x - d_L(x,y), y)| \geq \sigma_0 \end{cases}, \tag{12}$$

where d_i and d_r are the disparities of the left and right views respectively, and σ_0 is the tolerance for detecting the wrong disparity. To correct the mismatched pixels with $L(x, y) = 0$, there are several disparity refinement methods [26–31]. Usually, we can classify the mismatching pixels into large and small hole regions. For small hole regions, the background filling algorithm is used to improve the rough disparity map. For big hole regions, the four-step hole filling method can search the nearest reliable pixel in neighboring regions [31].

2.2 Simple linear iterative clustering

It is noted that the disparity map will have same disparity values in an object. In order to correctly estimate the disparity, the precise segmentation of the objects will help to improve accurate performance. With precise object boundaries, we could use them to improve the estimation of disparity map. It is noted that the precise object segmentation is computation consuming processes for left and right images. However, for stereo matching, we only need to perform a localized segmentation in small regions. In other words, we only need to identify the superpixels, which are collections of adjacent and homogeneous pixels of the images. The superpixel, as a segment, provides more structure information than a single pixel.

In this paper, we adopt simple linear iterative clustering (SLIC) [32], which adapts k-means clustering method to efficiently group the superpixels. The SLIC method with five-dimension space of $\{l_i, a_i, b_i, x_i, y_i\}$ localizes the ith pixel search range to an area

associated with the cluster center to reduce the computation, where (l, a, b) is the pixel color vector defined in CIELAB color space and (x, y) is the pixel position. The SLIC algorithm, which measures the distance between the ith pixel to the cluster center, considers both color similarity and spatial proximity, which are respectively denoted as

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}, \tag{13}$$

and

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}, \tag{14}$$

where $\{l_k, a_k, b_k, x_k, y_k\}$ is the cluster center. The k-means clustering is then applied to achieve superpixel segmentation. With the SLIC method, the utilization of segmentation results could provide more matching information for local stereo matching algorithms.

3 The proposed stereo matching system

Comparing to the traditional method depicted in Fig. 1, the corresponding functional diagram of the proposed stereo matching system is shown in Fig. 3, which uses SLIC-based cost aggregation for estimating the accurate left and right depth maps.

To exhibit the usages of SLIC segmentation information, Fig. 4 shows two innovated kernels: the adaptive stereo matching computation unit first estimates the left and right raw disparity maps while the dual-path refinement unit further enhances them to become accurate disparity maps. The descriptions of the kernels are addressed in the following subsections.

3.1 Adaptive stereo matching computation

Figure 5 shows the diagram of the proposed adaptive stereo matching computation unit, which includes the adaptive cost selection of gradient cost, census cost, or SAD cost, the SLIC-based cost aggregation with left and right SLIC segmentation information, and 2-level winner takes all to estimate the left and right raw disparity maps.

3.1.1 Adaptive cost selection

To estimate the similarity between the pixels in the left and its corresponding right image, the initial cost computation is necessary. First, we detect the edge regions in color image by using Sobel operator such that we can classify the pixels into edge region or non-edge region. For edge regions, we will use gradient cost as the initial cost. For non-edge region, we further classify it as a smooth or non-smooth region. Here, we utilize the cross-based window [22] to identify the smooth region. The criterion for the adaptive cost selection of SAD, gradient, or census cost is shown in Fig. 6.

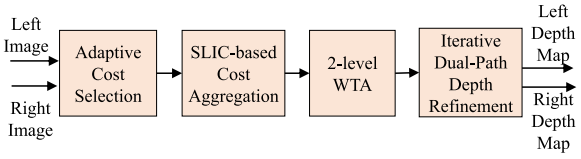

Fig. 3 Flowchart of the proposed stereo matching system

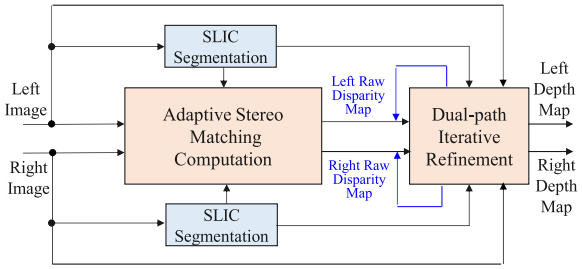

Fig. 4 Kernels of the proposed stereo matching system

If the pixel is classified in the edge region, the gradient similarity as stated in (2) is used since the variation in color image is large. If the pixel is classified as the non-edge region, we will use cross-based window to further verify whether the pixel lies on smooth region or not. To find a smooth plane, we calculated the cross-based plane as

$$r* = \max_{r\in[1,L]} \left(r \prod_{i\in[1,L]} \delta(p,p_i) \right), \tag{15}$$

where r^* denotes the largest left span in one direction and the indicator function is defined by

$$\delta(p,p_i) = \begin{cases} 1, \max\limits_{c\in\{R,G,B\}} (|I_c(p) - I_c(p_i)|) \le \tau_1 \\ 0, \text{otherwise} \end{cases}, \tag{16}$$

to evaluate the color similarity of pixels. In (15) and (16), p_i is the pixel extended in the direction. Once the largest span arm r^* is derived, we define the left arm length h_p^- = max (r^*, 1). Similarly, we can find the other three directions to obtain the arm lengths as $\{h_p^-, h_p^+, v_p^-, v_p^+\}$ for the pixel p. The two orthogonal cross segments are given as

$$H(p) = \left\{ (x,y) \middle| x\in\left[x_p - h_p^-, x_p + h_p^+\right], y = y_p \right\}, \tag{17}$$

$$V(p) = \left\{ (x,y) \middle| x = x_p, y\in\left[y_p - v_p^-, y_p + v_p^+\right] \right\}, \tag{18}$$

After computation of pixel-wise cross decision results, we can obtain a shape-adaptive full support region $U(p)$ for the pixel at p. The support region is an area integral of multiple segments $H(p)$ and is defined as

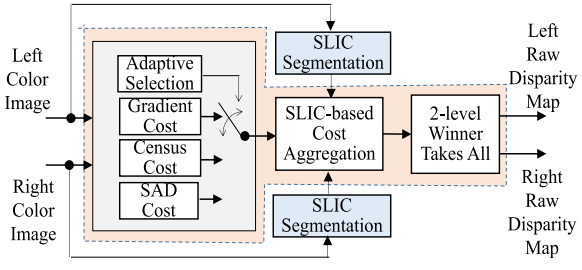

Fig. 5 Diagram of the adaptive stereo matching computation unit to estimate left and right raw disparity maps

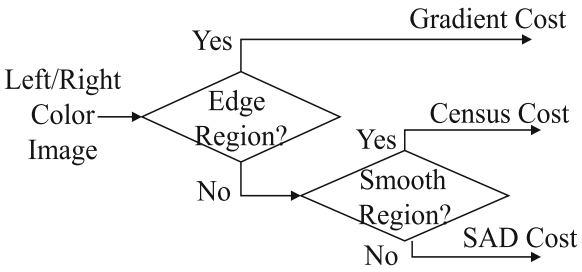

Fig. 6 Flow chart of adaptive cost selection strategy

$$U(p) = \bigcup_{p_v \in V(p)} H(p_v), \tag{19}$$

where p_v is a support pixel located on the vertical segment $V(p)$. If the area of the cross-based plane is more than 80% of the intact window, we classify the pixel lies on a smooth region. Once the pixel is classified in the smooth region, we use the census cost defined in (5) for stereo matching. On the contrary, if we classify the pixel in non-smooth region, the SAD cost as stated in (1) will be used. For stereo matching cost, Fig. 7 shows the results of raw disparity map achieved by using the direct combined initial cost and the adaptive selected initial cost. In consideration of different texture features, the proposed adaptive cost selection achieves better raw disparity maps in both complex texture regions and smooth regions.

3.2 Cost aggregation with SLIC-based ASW

For cost aggregation, we use adaptive support weights (ASWs), which are determined by SLIC segmentation information [32]. For each segmented superpixel, the aggregated cost for the pixels in the same segment should give them higher weights. The aggregation weights in the superpixel concept will be better than the geometry and color similarities in pixel-by-pixel fashions. First, we segment the color image into K levels by the SLIC segmentation algorithm. The segments in a higher level will have a more complex segmentation map. From low to high levels, if the neighboring pixels and the center

Fig. 7 Disparity maps of Tsukuba and Teddy test images achieved by **a** left color images, **b** direct combined cost, and **c** adaptive cost selection

pixel are in the same segmented superpixels, these pixels, which are prone to have higher similarity, should be given with higher weights. Figure 8 shows the result of different level segmentation images.

For K-level system, the proposed SLIC-based adaptive weight is given as

$$w(p,q) = \begin{cases} \exp(-N_s(p,q)/r) & ,\text{if } N_s(p,q) \le 0.5K; \\ \exp(-\Delta c(I_c(p), I_c(q))/r_c) & ,\text{if } N_s(p,q) > 0.5K, \end{cases}, \tag{20}$$

where $N_s(p, q)$ denotes the segmentation dissimilarity defined as

$$N_s(p,q) = \sum_{k=1}^{K} T\left[S_p^k \neq S_q^k\right], \tag{21}$$

where T [·] is an indicator function whose value equals to 1 when p and q are not in the same segment at the kth level, and 0 otherwise. In (21), $S^k{}_p$ and $S^k{}_q$ are the segmentation labels of pixels p and q at the kth level, respectively. To avoid the ambiguity in the dissimilar pixels, we suggest the adaptive weight based on the color difference to increase the accuracy if the dissimilarity is over half of total levels. The SLIC-based adaptive weights help to obtain a more reasonable aggregation cost to improve the disparity estimation than the cost aggregation weights stated in (10).

It is noted that the proposed adaptive weights can reduce the distortion of the similar pixels and keep the sensitive in complex texture regions. If we only use segmentation similarity part, called SLIC-only, without adaptive weights controlled by color changes, the variations of weights cannot tell the detailed differences. Figure 9 shows two distributions of the adaptive weights along x-axis obtained by the proposed method (blue color) and the SLIC-only (red color). If their weights are the same, we show them with mixed (purple) color. Thus, the weights obtained by the SLIC-only are hard to separate the differences in complex region since they are nearly equal and of low values. As the results, the proposed adaptive weights obtained in (22) can successfully avoid the ambiguity conditions with large variation weights.

3.3 Two-level WTA strategy

Normally, the WTA strategy is used to select the disparity value with the minimum cost. However, there might exist over one disparity sharing the same minimum cost or have several similar minimum costs. In order to avoid inaccurate disparity decision, we modified WTA into two-level procedure. First, we check every pixel as

Fig. 8 Results of different levels of the segmented image

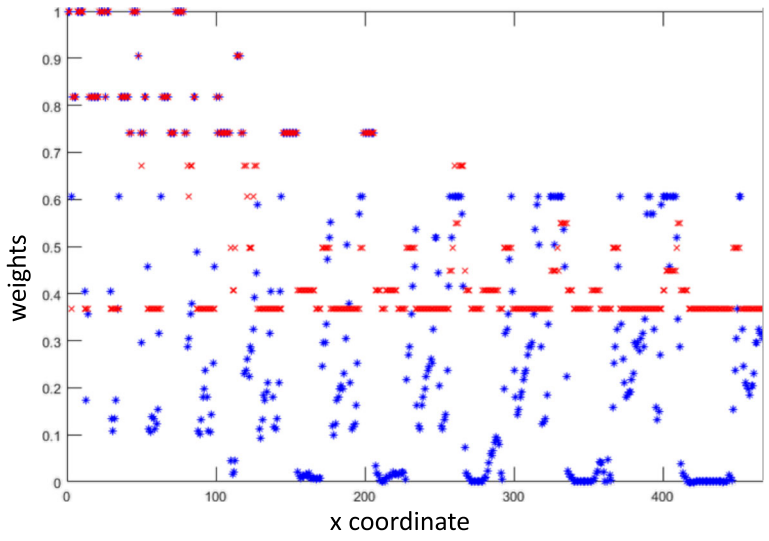

Fig. 9 Variations of the SLIC-only and the proposed adaptive aggregation weights

$$d(p) = \begin{cases} \arg\min_{d\in D} C_{agg}(p,d) & ,\text{if } N\left(\min_{d\in D} C_{agg}(p,d)\right) < 3 \\ 256 & ,\text{if } N\left(\min_{d\in D} C_{agg}(p,d)\right) \geq 3 \end{cases}, \tag{22}$$

where $N(\cdot)$ represents the number of disparities, which have the same minimum cost. If we have more than 3 candidates, which share the same minimum cost, we will replace $d(p)$ by 256 to label the pixel p as an unstable point. To deal with the unstable points, we use window-based histogram voting to select the correct disparity. For each pixel p, a histogram $H_p(d)$ of the stable points surrounding p in this window is created. The histogram bin with the highest value $d_v(p)$ is selected to replace the unstable point as

$$d_v(p) = \arg\max_{d\in D} H_p(d). \tag{23}$$

After the disparity of each pixel is found, we could adjust the scale of the disparities to generate raw depth maps. Generally, the left and right images will have slight intensity difference except the whole object is flat or perpendicular to the paired cameras. The minimum matching cost might not be able to find the correct matching point. With the proposed method, the truly disparity could be obtained more precisely.

3.4 Iterative dual-path depth refinement

Since the estimated raw disparity map usually contains some mismatches occurring near object boundaries and smooth regions. It is hard to reserve the shapes in these regions. Thus, we propose an iterative dual-path refinement algorithm to refine the raw disparity maps to obtain high precision depth maps and shape reserved.

To find the mismatched disparity, we first label the disparity map by the modified LRC as,

$$L(x,y) = \begin{cases} 1, & \text{if } d_l(x,y) = d_r(x - d_l(x,y), y) \cap \left\| I_{d_l} - I_{d_r(x-d_l)} \right\| \leq \tau_{AD}; \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

In (24), not only with disparity similarity, we further include the color tolerance to label the pixels. For $L(x, y) = 0$, the mismatched pixels are further categorized into two

types: small holes or big holes. If the mismatching region between the pixel in the target view is less than 2 pixels, we classify them as small holes. Otherwise, the other mismatched pixels are classified as big holes. Figure 10 shows the flow chart of proposed iterative dual-path refinement.

3.4.1 Small hole filling

Since the mismatching region contains small holes, the color image helps to find the accurate disparity by obtaining the texture information. Here, we utilize maximum-weighted candidate to find the correct disparity. With the image color similarity and spatial proximity in a correction window, we calculate the weight of each pixel as

$$\omega(p,q) = \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma c} + \frac{\Delta s_{pq}}{\gamma s}\right)\right), \tag{25}$$

where Δc_{pq} and Δs_{pq} are the color differences in RGB domain and spatial difference, respectively. We analyze the disparity distribution with the calculated weight. Under the disparity in the ascending order weighted histogram, the maximum corresponding disparity in the ordered histogram is the point of the final disparity, which is written as

$$d_{\text{out}}(p) = \left\{ d(q) \;\middle|\; \max_{q\in\Omega} \omega(p,q) \right\}, \tag{26}$$

where Ω is the correlation window region and d_{out} is the final refined disparity map.

3.4.2 Dual-path big hole filling

For big holes, finding the correct disparity in the surrounding pixels is not suitable in this circumstance. Here we should first classify the occlusion region into non-border occlusion and leftmost/rightmost border occlusion. Then, we designed two-path hole filling for both cases. For non-border regions, the holes, which are induced by the occlusion of the foreground objects, should be filled with the background disparity. On the other hand, the holes should be considered on the target color image only. The flow chart of occlusion region refinement in two paths is shown in Fig. 11, and the processing details are described as the following.

For non-border hole filling, we usually directly use the background information to fill the pixels with the mismatched disparity. To get more accurate disparity map, we make use of the similar pixels in background of the color image. First, we calculate the color

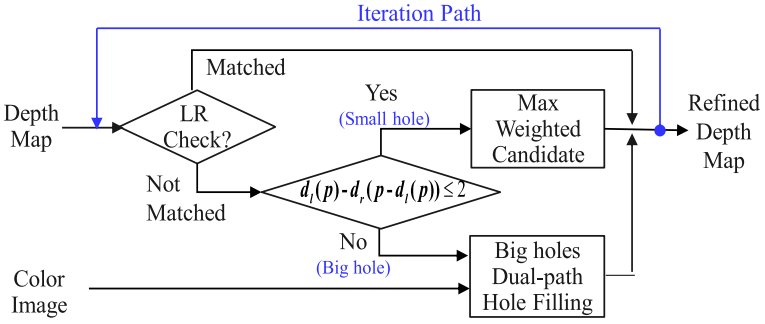

Fig. 10 Flow chart of iterative dual-path refinement

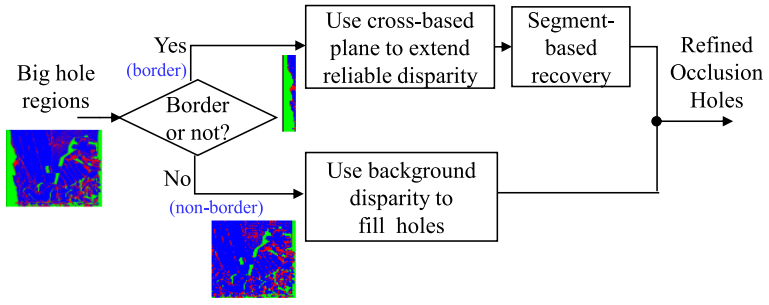

Fig. 11 Flow chart of dual-path refinement for big hole regions

similarity to find the most similar pixel on the same horizontal line among Q pixels to fill its corresponding disparity of the occluded pixel. After finding the similar pixel in the background (extended to the left side), we assign its corresponding disparity to the hole as

$$D_{oc} = \left\{ d(x-i) \middle| \arg\arg \min_{0 \le i \le Q} \Delta C(x, x-1) \right\}, \tag{27}$$

where $\Delta C(x, x - i)$ denotes the color similarity between the target hole at x and the horizontal-left background pixel at $x - i$. Though there are still some residual wrong disparities by the proposed non-border hole filling method, the problem can be solved by iterative steps. The illustrations of the regular background hole filling and the matched-color background hole filling are shown in Fig. 12 a and b, respectively. We did not fill the hole by the nearby background disparity (light blue) pixel. We filled the marched-color background disparity (yellow) pixel.

For the leftmost border regions in the target (left) disparity map, as shown in Fig. 13 a and b, we only can refer the target (left) color image to fill the holes of the target (left)

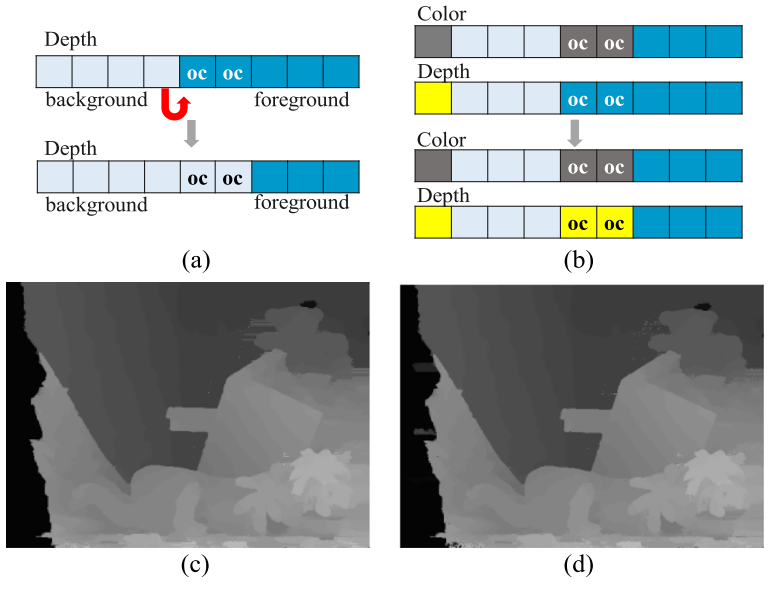

Fig. 12 Illustrations and results of the occlusion small-hole background filling methods. **a** Regular. **b** Matched-color. **c** Disparity map by regular. **d** Disparity map by matched-color

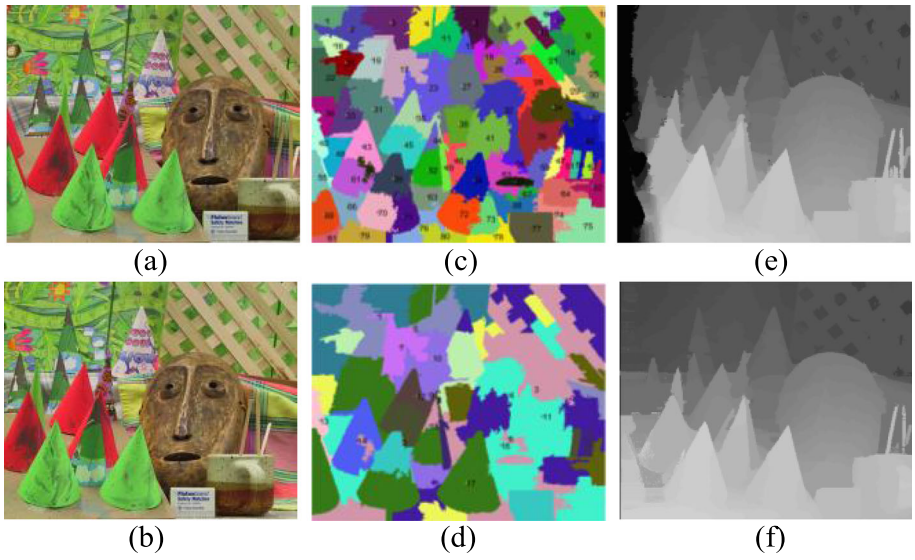

Fig. 13 Border disparity segment-based refinement. **a** Left color image. **b** Right color image. **c** Segmented superpixels of left image. **d** Merged megapixels of left image. **e** Left raw disparity map. **f** Left refined disparity map

disparity map since we cannot find any matching information from the reference (right) view. The object in the leftmost region of the right image totally disappears. We do not have any clue to find the corresponding disparity for unknown regions. Thus, we only can use the leftmost color image to infer the holes as possible. Fortunately, we have computed SLIC segments for determination of ASWs to the color image as shown in Fig. 13c, which shows the localized superpixels. We can use the concept that the pixel in the same superpixel should have the same depth value. For better inferences, we could associate the localized SLIC superpixels for border big-hole filling as the following procedures: First, we could merge the localized superpixels, which have similar texture color information, as shown in Fig. 13d, to gather some superpixels into larger megapixels, which are treated as the object-like segments; secondly, we horizontally extend the known and reliable disparity leftward to all the hole pixels, which share the same megapixel as possible. We can obtain some filled megapixels in this step.

Thirdly, we perform disparity histogram voting for those isolated megapixels, which do not contain any known disparity. Starting from the lowest pixel of the isolated megapixel, we choose the disparity from the largest disparity histogram of the filled megapixel, which is next to the current megapixel. Finally, the hole regions in the border can be reproduced with clear objects and their edges. The left refined disparity map is shown in Fig. 13f.

Table 1 Image information of Middlebury 2005

Images	Image resolutions	Disparity level
Cones	450 × 375	60
Teddy	450 × 375	60
Tsukuba	384 × 288	16
Venus	434 × 383	20

Table 2 Image information of Middlebury 2014

Images	Image resolutions	Disparity level
Adirondack	718 × 496	73
ArtL	347 × 277	64
Jadeplant	659 × 497	160
Motorcycle	741 × 497	70
MotorcycleE	741 × 497	70
Piano	707 × 481	65
PianoL	707 × 481	65
Pipes	735 × 485	75
Playroom	699 × 476	83
Playtable	680 × 463	73
PlaytableP	681 × 462	73
Recycle	720 × 486	65
Shelves	738 × 497	60
Teddy	450 × 375	64
Vintage	722 × 480	190

4 Results and discussion

The proposed stereo matching system was implemented with MATLAB R2016a and tested on an Intel Core i5-8400 PC and 16 GB RAM. The experimental evaluation is performed by using 2003 [33], 2005 [34], and 2014 [35] datasets created in Middlebury. The testing images that include Cones, Teddy, Tsukuba, and Venus are shown in Table 1 while the test images with higher resolutions and higher disparity levels are exhibited in Table 2.

4.1 Results achieved by the proposed system

As shown in Fig. 14, the raw and refined disparity maps achieved by the proposed adaptive stereo matching and dual-path refinement methods for Cones, Teddy, Tsukuba, and Venus test images are exhibited in the first and second rows, respectively.

Fig. 14 Disparity maps for Cones, Teddy, Tsukuba, and Venus test images achieved by **a** raw disparity by the proposed adaptive stereo matching, and **b** refined disparity after dual-path refinement

Table 3 Error rate (%) of the proposed and other stereo matching algorithms

Methods		Sun [31]	Hsieh [29]	Kuo et al. [27]	E-SM [37]	S-ASW [23]	Proposed
Cones	Non-occ	6.69	5.62	4.44	15.64	7.44	5.59
	All	13.72	13.23	10.96	24.6	13.49	12.06
	Disc	18.70	14.36	10.22	27.22	16.79	13.24
Teddy	Non-occ	10.57	8.75	9.39	20.84	8.73	9.57
	All	17.96	16.23	15.52	28.83	14.78	15.50
	Disc	23.60	26.32	19.16	34.81	23.57	25.08
Tsukuba	Non-occ	6.70	4.26	4.38	5.15	2.59	2.79
	All	8.11	5.18	5.35	7.05	2.89	3.05
	Disc	19.67	17.42	17.12	19.93	13.26	13.93
Venus	Non-occ	1.54	1.98	4.15	5.49	0.48	0.38
	All	2.08	2.43	5.09	7.01	0.68	0.64
	Disc	12.96	6.45	10.71	30.02	4.21	4.88
Average		11.86	9.21	9.23	10.40	9.08	**8.89**

4.2 Comparisons with other approaches

For performance evaluations, we compare the proposed method to other stereo matching algorithms. The compared methods include adaptive support weight (ASW) [21], segmentation-based adaptive support weight (S-ASW) [23], plant leaf stereo matching (LP-SM) [36], edge-based stereo matching method (E-SM) [37], stereo matching implemented on GPU platform [31], AdaStereo [38], comparative evaluation of SGM variants for dense stereo matching (tMGM) [39], learning-based disparity estimation (iResNet) [40], and DeepPruner [41] methods. Tables 3 and 4 show that the performance of the proposed multi-scale ASW is superior to traditional ASW and other methods. Table 4 shows we have better performance than some deep learning-based methods in training set, even the training set is more beneficial to deep learning. Though the average error

Table 4 Error rate (%) of the proposed and other stereo matching algorithms

Methods		AdaStereo [38]	tMGM [39]	iResNet [40]	DeepPruner [41]	Proposed
Adirondack	Non-occ	28.1	29.9	25.1	36.9	16.6
ArtL	Non-occ	11.0	11.9	20.2	37.4	30.9
Jadeplant	Non-occ	44.4	28.4	47.9	56.5	21.4
Motorcycle	Non-occ	37.0	16.1	35.6	41.2	11.5
MotorcycleE	Non-occ	37.9	16.1	36.2	41.7	35.5
Piano	Non-occ	39.5	25.8	37.4	37.9	14.2
PianoL	Non-occ	62.2	36.3	59.8	47.4	43.2
Pipes	Non-occ	36.7	14.2	34.5	47.5	16.3
Playroom	Non-occ	49.8	38.5	46.1	52.9	34.4
Playtable	Non-occ	51.4	30.4	40.2	44.9	21.7
PlaytableP	Non-occ	37.6	25.4	25.3	38.6	15.3
Recycle	Non-occ	31.0	31.0	35.0	34.9	10.0
Shelves	Non-occ	63.8	52.7	59.6	52.7	15.5
Teddy	Non-occ	14.5	11.7	15.0	22.9	11.9
Vintage	Non-occ	51.8	52.8	47.7	41.7	30.1
Average		36.6	25.3	35.1	41.2	**21.9**

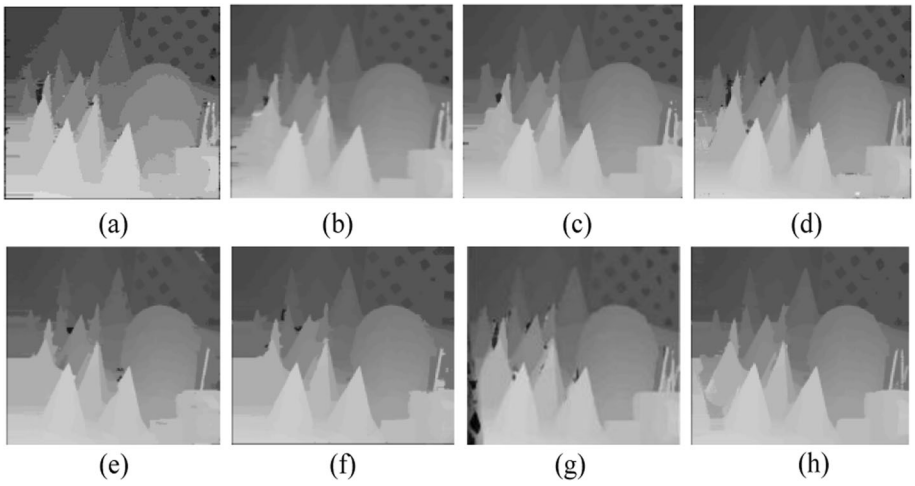

Fig. 15 Estimated disparity maps of Cones achieved by **a** Kuo et al. [27], **b** Kuo [28], **c** Hsieh [29], **d** Sun [31], **e** ASW [21], **f** S-ASW [23], **g** LF-SM [36], and **h** the proposed method

rate is slightly lower than S-ASW, our method utilizes more information from segmentation instead of only assign weight to 1 with the same segment. According to the refinement steps, the edge areas of the depth maps can be reasonably reconstructed. With the proposed algorithm, the disparity maps show accurate, which helps to improve the performance in the DIBR system for multi-view synthesis [42]. Figures 15, 16, 17 and 18 show the results achieve by the referenced methods for Cones, Teddy, Tsukuba, and Venus images, respectively.

.

5 Conclusions

In this paper, we proposed a segment-based adaptive stereo matching algorithm and a dual-path disparity segment-based refinement method. The former can provide a reasonable good raw disparity map, and the latter can effectively enhance the raw disparity map into high-quality ones. The contributions of the proposed method include

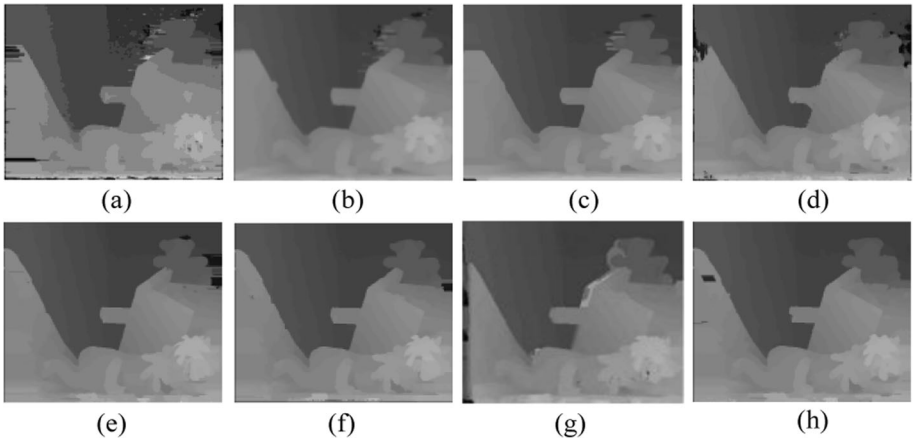

Fig. 16 Estimated disparity maps of Teddy achieved by **a** Kuo et al. [27], **b** Kuo [28], **c** Hsieh [29], **d** Sun [31], **e** ASW [21], **f** S-ASW [23], **g** LF-SM [36], and **h** the proposed methods

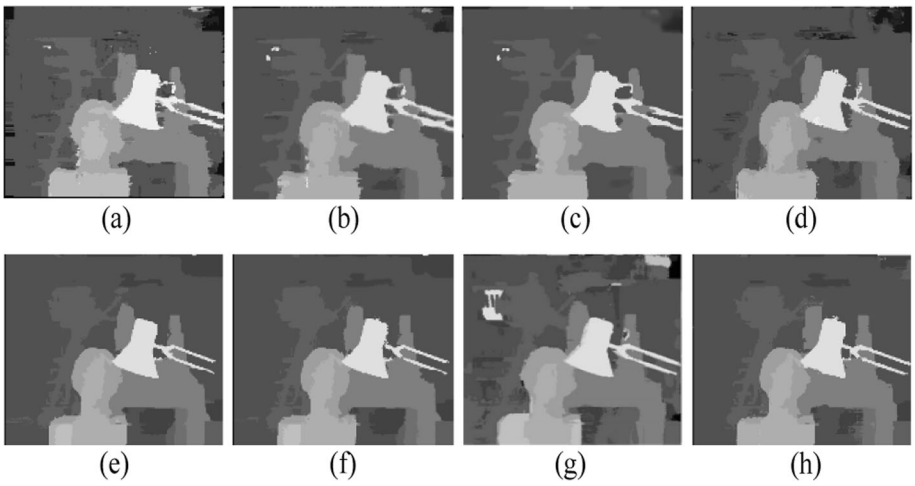

Fig. 17 Estimated disparity maps of Tsukuba achieved by **a** Kuo et al. [27], **b** Kuo [28], **c** Hsieh [29], **d** Sun [31], **e** ASW [21], **f** S-ASW [23], **g** LF-SM [36], and **h** the proposed methods

adaptive cost selection, the segment-based adaptive weights for cost aggregation, two-level WTA strategy, and dual-path depth refinement. For small holes, the depth refinement uses maximum-weighted candidate for the best filling process. For non-border big holes, the background filling strategy is adopted by consideration of color and proximity information. And for border holes, the megapixel-based filling process is suggested to achieve better results. The proposed stereo matching system tested on Middlebury stereo datasets shows the best performances among all compared methods. Especially in the edge areas of the depth maps, it can reasonably reconstruct depth values of the objects. The experimental results show that the proposed system can reach high-quality depth maps for 3D video broadcasting with 3D-HEVC [43, 44] and CTDP-HEVC [45, 46] formats. Comparing with the deep-learning methods, the proposed system can be applied to various databases. As to learning-based approaches with

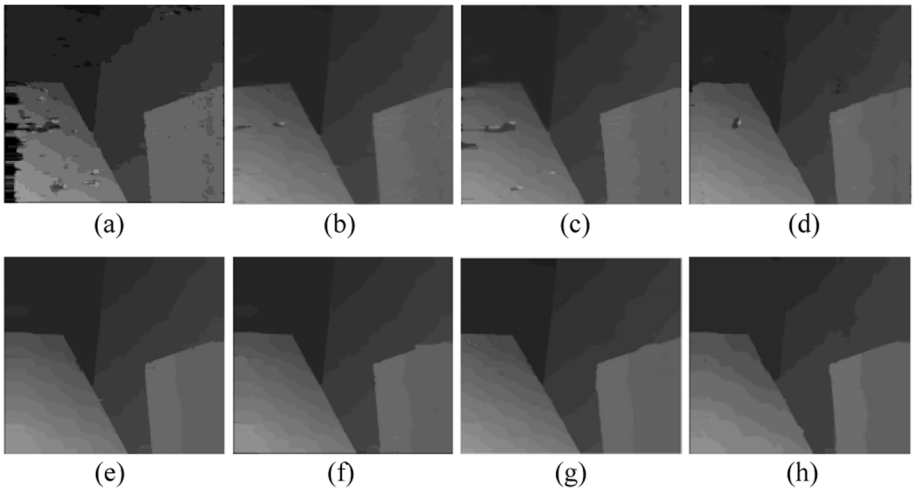

Fig. 18 Estimated disparity maps of Venus achieved by **a** Kuo et al. [27], **b** Kuo [28], **c** Hsieh [29], **d** Sun [31], **e** ASW [21], **f** S-ASW [23], **g** LF-SM [36], and **h** the proposed methods

convolutional neural networks, they have problems in data dependencies and are easily blurred at depth edges because of the designs of the loss functions.

Abbreviations

3D: Three-dimension; DIBR: Depth image-based rendering; SAD: Summation of absolute differences; WTA: Winner-take-all; LRC: Left right consistency check; SLIC: Simple linear iterative clustering; ASWs: Adaptive support weights; S-ASW: Segmentation-based adaptive support weight; LP-SM: An improved stereo matching algorithm applied to 3D visualization of plant leaf; E-SM: Variable window size for stereo image matching based on edge information; tMGM: SGM variants for dense stereo matching; iResNet: Learning for disparity estimation; CTDP: Centralized texture depth packing

Acknowledgements

The authors deeply thank the Editor and anonymous reviewers who have spent their valuable time to review this paper and give constructive suggestions for improvements of formatting and readability of the paper.

Authors' contributions

C.-S. Huang carried out image processing studies, participated in the proposed system, and drafted the manuscript. Y.-H. Huang carried out the software simulations and adjustment parameters. D.-Y. Chan and J.-F. Yang conceived of the study and participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST 106-2221-E-006 -038 -MY3.

Availability of data and materials

All the data and material are from Middlebury datasets, which have been mentioned in the references.

Competing interests

The authors declare that they have no competing financial interests.

Author details

[1]Department of Electrical Engineering, National Cheng Kung University, 1, University Rd, Tainan City, Taiwan. [2]Department of Computer Science and Information Engineering, National Chia-Yi University, Chia-Yi, Taiwan.

Received: 29 February 2020 Accepted: 17 August 2020
Published online: 07 September 2020

References

1. R. Kaiser, D. Schatsky, For more companies, new ways of seeing – Momentum is building for augmented and virtual reality in the enterprise. Deloitte, Insights **5** (2017)
2. L. Zhang, Fast stereo matching algorithm for intermediate view reconstruction of stereoscopic television images, IEEE Trans Circuits Syst Video Technol, **16**(10), 1259 – 1270, Oct. (2006).
3. S. Carmichael, Using 3D immersive technologies for organizational development and collaboration. Organizational Dynamics Working Papers, University of Pennsylvania, May **1** (2011)
4. KPMG – FOCCI, The future: now streaming, Indian Media and Entertainment Industrial Report, (2016).
5. J. H. Joung, K. H. An, J. W. Kang, M. J. Chung and W. Yu, 3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, USA (2009)
6. S. Kriegel, C. Rink, T. Bodenmüller and M. Suppa, Efficient next-best-*scan* planning for autonomous 3D surface reconstruction of unknown objects, J. Real-Time Image Proc., **10**(4), 611-631, Dec. (2015).
7. X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3D object proposals for accurate object class detection, Proc. of Advances in Neural Information Processing Systems 28 (NIPS), (2015)
8. S. Song and J. Xiao, Sliding shapes for 3D object detection in depth images, Proc. of European Conference on Computer Vision. Pp.634-651, (2014).
9. E. Zappa, P. Mazzoleni, Y. Hai, Stereoscopy based 3D face recognition system. Proc Comput Sci. **1**(1), 2529–2538 (2010)
10. S.C. Chan, H. Shum, K. Ng, Image-based rendering and synthesis. IEEE Signal Process. Mag. **24**(6), 22–33 (2007)
11. I.P. Howard, B.J. Rogers, *Binocular Vision and Stereopsis* (Oxford University Press, USA, 1995)
12. Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Machine Intell **23**(11), 1222–1239 (2001)
13. X. Sun, X. Mei, S. Jiao, M. Zhou and H. Wang, Stereo matching with reliable disparity propagation, Proc. of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, 2011, pp. 132-139. (2011)
14. J. Sun, N.-N. Zheng and H.-Y. Shum, Stereo matching using belief propagation, IEEE Trans Pattern Anal Machine Intell, **25**(7), 787-800, July (2003).
15. O. Veksler, Stereo correspondence by dynamic programming on a tree, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, (2005).
16. W. Luo, A. G. Schwing and R. Urtasun, Efficient deep learning for stereo matching, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 5695-5703. (2016)
17. M. Humenberger, T. Engelke, and W. Kubinger, A census-based stereo vison algorithm using modified semi-global matching and plane fitting to improve matching quality, Proc. of IEEE Computer Vision Patter Recognition Conf., pp. 77-84, (2010).

18. N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, T.-S. Chang, Algorithm and architecture of disparity estimation with mini-census adaptive support weight, IEEE Trans Circuits Syst Video Technol, **20**(6), 792 – 805, June (2010).
19. T. Chen and W. Li, Stereo matching algorithm based on adaptive weight and local entropy, Proc. of the 9th International Conference on Modelling, Identification and Control (ICMIC), Kunming, 2017, pp. 630-635. (2017)
20. O. Veksler, Fast variable window for stereo correspondence using integral images, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 2003, pp. I-I. (2003)
21. K. J. Yoon and I. S. Kweon, Adaptive support-weight approach for correspondence search, IEEE Trans Pattern Anal Machine Intell, **28**(4), 650-656, April (2006).
22. K. Zhang, J. Lu and G. Lafruit, Cross-based local stereo matching using orthogonal integral images, IEEE Trans Circuits Syst Video Technol, **19**(7), 1073-1079, July (2009).
23. F. Tombari, S. Mattoccia, L. Di Stefano, Segmentation-Based Adaptive Support for Accurate Stereo Correspondence in *Lecture Notes in Computer Science*, Berlin, Germany: Springer, 4872, pp. 427-438, Dec. (2007).
24. D. Chang, S. Wu, H. Hou, L. Chen, Accurate and fast segment-based cost aggregation algorithm for stereo matching. Proc. of IEEE 19th International Workshop on Multimedia Signal Processing, 1–6 (2017, 2017)
25. H. Zhu, J. Yin, D. Yuan, SVCV: Segmentation volume combined with cost volume for stereo matching. IET Comput. Vis. **11**(8), 733–743 (2017)
26. S. B. Kang, R. Szeliski and J. Chai, Handling occlusions in dense multi-view stereo, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, HI, USA, 2001, pp. I-I, (2001)
27. P.C. Kuo, K.L. Lo, H.K. Tseng, K.T. Lee, B.D. Liu, J.F. Yang, Stereoview to multiview conversion architecture for auto-stereoscopic 3D displays. IEEE Trans Circuits Syst Video Technol **28**(11), 3274–3287 (2017)
28. H. T. Kuo, VLSI Implementation of real-time stereo matching and centralized texture depth packing for 3D video broadcasting, M.S. Thesis, National Cheng Kung University, Tainan, Taiwan, July (2017).
29. C. L. Hsieh, A two-view to multi-view conversion system and its VLSI implementation for 3D displays, M. S. Thesis, National Cheng Kung University, Tainan, Taiwan, July (2017).
30. A. Emlek, M. Peker and K. F. Dilaver, Variable window size for stereo image matching based on edge information, *Proc. of International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, 2017, pp. 1-4. (2017)
31. T. Y. Sun, Stereo matching and depth refinement on GPU platform, M. S. Thesis, National Cheng Kung University, Tainan, Taiwan, July (2018).
32. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Machine Intell **34**(11), 2274–2282 (2012)
33. D. Scharstein and R. Szeliski, High-accuracy stereo depth maps using structured light, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, Madison, WI, pp. 195-202 June 2003.
34. D. Scharstein and C. Pal, Learning conditional random fields for stereo, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR 2007), Minneapolis, MN, Jun, (2007).
35. D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, P. Westling, *High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth, German Conference on Pattern Recognition (GCPR 2014)* (Münster, Germany, Sep, 2014)
36. Liu, Zhi-chao, Li-hong Xu, and Chao-feng Lin. An improved stereo matching algorithm applied to 3D visualization of plant leaf. *2015 8th International Symposium on Computational Intelligence and Design* (ISCID). **2**. IEEE, (2015).
37. Emlek, A., Peker, M., & Dilaver, K. F., Variable window size for stereo image matching based on edge information, *2017 International Artificial Intelligence and Data Processing Symposium* (IDAP), IEEE, pp. 1–4, September (2017).
38. Song, X., Yang, G., Zhu, X., Zhou, H., Wang, Z., & Shi, J, AdaStereo: a simple and efficient approach for adaptive stereo matching, *arXiv preprint arXiv*:2004.04627. (2020)
39. Patil, Sonali, Tanmay Prakash, Bharath Comandur, and Avinash Kak., A comparative evaluation of SGM variants (including a new variant, tMGM) for dense stereo matching, *arXiv preprint arXiv*:1911.09800, (2019).
40. Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., ... & Zhang, J, Learning for disparity estimation through feature constancy. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2811-2820, (2018).
41. Duggal, Shivam, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun, DeepPruner: learning efficient stereo matching via differentiable PatchMatch., In Proceedings of the IEEE International Conference on Computer Vision, pp. 4384-4393. (2019).
42. K. J. Hsu, GPU implementation for centralized texture depth depacking and depth image-based rendering, M. S. Thesis, National Cheng Kung University, Tainan, Taiwan, July (2017).
43. G. Tech, K. Wegner, Y. Chen, and S. Yea, 3D HEVC test model 3. Document: JCT3VC1005. Draft 3 of 3D-HEVC Test Model Description. Geneva, (2013).
44. D. Rusanovskyy, K. Müller, and A. Vetro, Common test *conditions of 3DV core experiments*, joint collaborative team on 3D video coding extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Document no. JC3VC-E1100, Vienna, Aug. (2013).
45. J.-F. Yang, K.-T. Lee, G.-C. Chen, W.-J. Yang and Lu Yu, A YCbCr color depth packing method and its extension for 3D video broadcasting services, IEEE Trans. on Circuits and Systems for Video Technology, ISSN: 1051-8215, Online ISSN: 1558-2205 Digital Object Identifier: https://doi.org/10.1109/TCSVT.2019.29342541, pp.1-11. (2019)
46. W.-J. Yang, J.-F. Yang, G.-C. Chen, P.-C. Chung, M.F. Chung, An assigned color depth packing method with centralized texture depth packing formats for 3D VR broadcasting services. IEEE J Emerg Selected Topics Circuits Systems **9**(1), 122–132 (2019)

Publisher's Note