Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Near-real-time stereo matching method using both cross-based support regions in stereo views

Sangyoon Lee
Hyunki Hong

SPIE.

Sangyoon Lee, Hyunki Hong, “Near-real-time stereo matching method using both cross-based support regions in stereo views,” *Opt. Eng.* **57**(2), 023103 (2018), doi: 10.1117/1.OE.57.2.023103.

Near-real-time stereo matching method using both cross-based support regions in stereo views

Sangyoon Lee and Hyunki Hong*
Chung-Ang University, School of Integrative Engineering, Dongjak-ku, Seoul, Republic of Korea

Abstract. This paper presents a near-real-time stereo matching method using both cross-based support regions in stereo views. By applying the logical AND operator to the cross-based support region in the reference image and target image, we can obtain an intersection support region, which is used as an adaptive matching window. The proposed method aggregates absolute difference estimates in the intersection support region, which are combined with the census transform results. The census transform with a fixed window size and shape is applied, and only the resultant binary code of the pixel in the intersection support region is used. From Middlebury images and their ground truth disparity maps, we compute the area similarity ratio of support regions in stereo views. Then, a conditional probability of observing a correct disparity estimate with respect to the area similarity ratio is examined. By taking a natural logarithm of the probability, a relative reliability weight about the area similarity of support regions is obtained. The initial matching cost is then combined with the reliability weight to obtain the final cost, and the disparity with the minimum cost is chosen as the final disparity estimate. Experimental results demonstrate that the proposed method can estimate accurate disparity maps. [DOI: 10.1117/1.OE.57.2.023103]

Keywords: stereo matching; disparity map; support region; adaptive matching window; conditional probability.

Paper 170837 received Jun. 2, 2017; accepted for publication Jan. 19, 2018; published online Feb. 14, 2018.

1 Introduction

Stereo vision applications based on stereo matching are becoming increasingly common, ranging from mobile robotics to driver assistance system. The goal of stereo matching is to determine a precise disparity, which indicates the difference in the location of the corresponding pixels. The corresponding pixels between two images of the same scene are established based on similarity measures. Dense stereo matching to find the disparity for every pixel between two or more images has been actively researched for decades.[1–10]

Stereo matching algorithms are classified into global and local approaches.[1] Local methods utilize the color or intensity values within a finite support window to determine the disparity for each pixel. Global methods compute all disparities of an image simultaneously by optimizing a global energy function.[2–10] Global methods, which define a global energy function with a data term and smoothness term, help produce accurate disparity maps. To find the minimum global energy function, various global optimizers, such as dynamic programming (DP),[5,6] belief propagation,[7] and graph cuts,[8] have been proposed. Local algorithms select the potential disparity with the minimal matching cost at the pixel; hence, they are efficient and easy to implement.

Local stereo methods commonly use matching windows with a fixed size and shape, but the estimation results are greatly influenced by irrelevant pixels within the window considered. Improved local algorithms that reduce the error effects of irrelevant pixels can be divided into two categories.[3] Local stereo methods focus on selecting either the optimal window among predefined multiple windows or selecting point by point to adaptively support a window's size and shape.[9,10] However, building windows of various sizes and shapes adaptive to neighboring intensity distribution is time-consuming. Adaptive-weight methods assign different support weights to pixels in the given window by evaluating color similarity and geometric proximity,[2] but textureless regions, repeated patterns, and occlusion regions are not readily amenable to this solution.

This paper introduces an adaptive stereo matching method using the cross-based support regions in stereo views. The size and shape of the cross-based support region are chosen adaptively according to local color information and spatial distance. By applying the AND logical operator to the cross-based support region in both the reference image (left) and target image (right), an intersection support region can be obtained, which is then used as an adaptive matching window. The proposed method aggregates absolute difference (AD) estimates in the adaptive matching window, which are combined with the census transform results.

From the Middlebury reference images and their ground truth depth maps,[11] a conditional probability of observing a correct disparity estimate with respect to the similarity ratio of the areas in the cross-based support regions in stereo views can be calculated. When the cross-based support region in the target image is similar to that in the reference image, the area similarity ratio is close to 1, and a more accurate disparity value can be obtained. By taking the natural log probability, a relative reliability weight value based on the area similarity ratio is obtained. The initial matching cost is then combined with the reliability weight to obtain the final cost, and the disparity with the minimum cost is chosen as the final disparity estimate. Experimental results demonstrate that the proposed algorithm based on reliability weight

*Address all correspondence to: Hyunki Hong, E-mail: honghk@cau.ac.kr

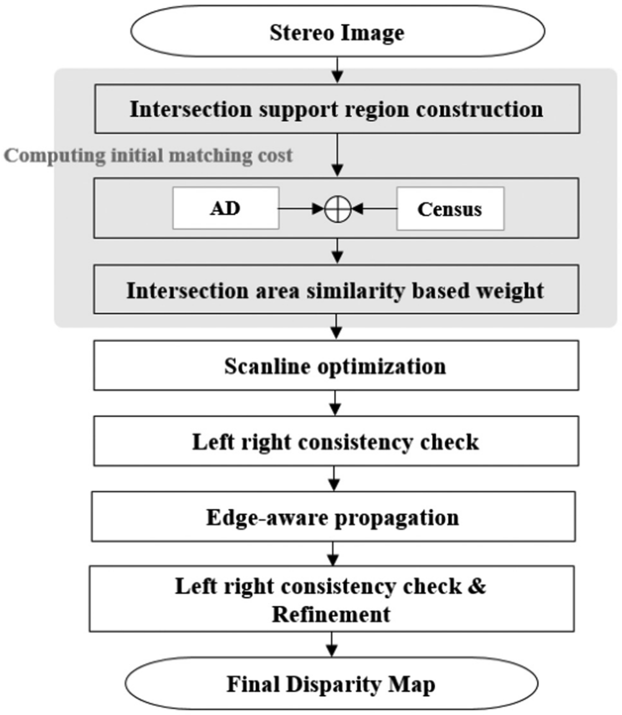

Stereo Image
Intersection support region construction
Computing initial matching cost
AD
Census
Intersection area similarity based weight
Scanline optimization
Left right consistency check
Edge-aware propagation
Left right consistency check &
Refinement
Final Disparity Map

Fig. 1 Proposed algorithm.

provides more accurate disparity estimates than previous methods.

The main contributions of this paper are twofold. First, using both support regions in stereo views enables a disparity map to be more accurately estimated. The intersection region of cross-based support regions in the reference image and target image is used as an adaptive matching window. Second, a reliability weight based on the area similarity ratio of both support regions in stereo views is introduced. The reliability weight represents the probability of correct disparity estimation with respect to the area similarity between stereo views. The shaded box in Fig. 1 represents the main components of our contribution.

2 Proposed Method

2.1 *Building Intersection Support Region*

In a textureless region, large matching windows to consider enough pixels are needed to overcome their erroneous matching costs, whereas in highly textured regions at depth discontinuities, smaller windows are needed to avoid oversmoothing. To address this problem, a cross-based support region construction method that can adaptively alter the window's shape and size is proposed,[3,4] which considers only the support region of the reference image. Another method considers cross-based support regions in both target and reference images, but the regions are used only for initial matching cost aggregation.[12] In this paper, an area where the cross-based support regions in both the target and reference images intersect is used as an adaptive support window for matching cost computation.

Reference 4 presented the enhanced construction rules to determine the shape of the cross-based support regions. Figure 2 shows how an upright cross with four arms for the anchor pixel $\mathbf{p} = (x, y)$ is constructed. Table 1 shows a pseudocode to determine the arm length l and endpoint, $\mathbf{p}_e$ of the horizontal and vertical directions with respect to $\mathbf{p}$. Here, $\mathbf{p}_n$ is the n'th pixel along the scanning direction of each arm: $\mathbf{p}_n = (x–n, y)$ in the left arm and $\mathbf{p}_n = (x + n, y)$ in

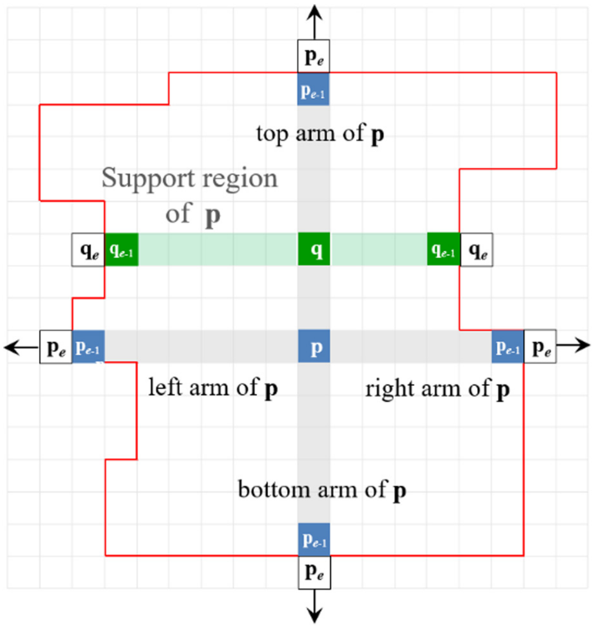


Fig. 2 Construction of upright cross with four arms for anchor pixel **p**.[3,4]

Table 1 Cross-based support region construction.

```
void function SupportRegionConstruction (anchor pixel p)
{
for ($i = 1$; $i <= 4$; $i++$)        % the scanline direction of the $i$'th arm of p
$l[i]$ = armLength (p);                 % Length of the $i$'th arm is saved as $l[i]$.
}
int function armLength (anchor pixel p)
{
  $n = 1$;
  for (; $n \le L_1$; $n++$) {
    if (($D_c(\mathbf{p}_n, \mathbf{p}) < \tau_1$) && ($D_c(\mathbf{p}_n, \mathbf{p}_{n-1}) < \tau_1$)) {
      if ($n > L_2$) {
        if ($D_c(\mathbf{p}_n, \mathbf{p}) < \tau_2$) continue;
        else break;
      }
    } else break;
  }
  return $n$;
}
```

the right arm, and the maximum number of $\mathbf{p}_n$ is set to L_1. In Table 1, we examine whether the color differences $D_c(\mathbf{p}_n, \mathbf{p})$ and $D_c(\mathbf{p}_n, \mathbf{p}_{n-1})$ are lower than τ_1 along the scanning direction of each arm. The color difference $D_c(\mathbf{p}_n, \mathbf{p})$ is defined as the maximum AD between $\mathbf{p}_n$ and $\mathbf{p}$ in RGB channels: $D_c(\mathbf{p}_n, \mathbf{p}) = \max_{i=R,G,B}|I_i(\mathbf{p}_n) - I_i(\mathbf{p})|$. Two successive pixels ($\mathbf{p}_n$ and its predecessor $\mathbf{p}_{n-1}$) are examined so that the arm will not go beyond the edges of the image. When the arm length is longer than L_2 ($n > L_2$), the lower color threshold value τ_2 ($\tau_2 < \tau_1$) is used in the color difference computation to control the arm length more flexibly. In this paper, τ_1, L_1, τ_2, and L_2 are the experimentally preset threshold values 27, 21, 15, and 13, respectively. Then, the support region of pixel $\mathbf{p}$ is modeled by merging the horizontal arms of all the pixels (for example, green colored row of $\mathbf{q}$ in Fig. 2) lying on the vertical arms of pixel $\mathbf{p}$. Each horizontal line (left arm and right arm) of $\mathbf{q}$ is examined as shown in Table 1, and the arm length l and endpoint $\mathbf{q}_e$ are determined. Since the color distribution is different for each row of pixel $\mathbf{q}$ lying on the vertical arms of $\mathbf{p}$, the support region of $\mathbf{p}$ is not rectangular.

In the AD method, the matching cost is computed as the AD in color/intensity between corresponding pixels.[1] The matching costs in the support region are aggregated within two passes along the horizontal and vertical directions. In the first pass, the matching costs are aggregated from the endpoint's predecessor $\mathbf{q}_{e-1}$ of the left arm of any pixel $\mathbf{q}$ to the endpoint's predecessor of the right arm. In other words, the horizontal summation of matching costs is performed on every pixel $\mathbf{q}$ lying on the vertical arms of pixel $\mathbf{p}$. Then, the intermediate results of $\mathbf{q}_s$ on the vertical arms are aggregated vertically to obtain the final cost. Both passes can be efficiently computed with one-dimensional integral images.[3,4]

Given a pixel $\mathbf{p} = (x, y)$ in the reference image, our goal is to establish its corresponding $\mathbf{pd} = (x\text{–}d, y)$ in the target image accurately. Since the stereo rig is assumed to be rectified, we only examined a horizontal translation. Here, we can acquire a support region $\mathrm{SR}(\mathbf{p})$ in the reference image and the support regions $SR\prime(\mathbf{pd})$ in the target image along a candidate disparity level d. Using the logical AND operation of the cross-based support region in the reference image and target image, the support region with the same size and shape, called the intersection support region, can be obtained.

Assuming that neighboring pixels with similar colors have similar disparities, previous methods built cross-based support regions for initial matching cost aggregation.[3,4] However, when the intensity distribution is complex, constructing the cross-based support regions of a surface with the same depth information is difficult. Furthermore, a support region of insufficient size may be built in this case. If the total arm length in both horizontal directions is less than five pixels, the proposed method sets the length of the support region to five pixels to consider the minimum neighborhood region.

2.2 *Computing Initial Matching Cost*

In the cross-based support region method,[4] the initial matching cost $C(\mathbf{p}, d)$ is computed by combining the AD measure and census transform in an exponential function. More specifically, given two corresponding pixels $\mathbf{p}$ and $\mathbf{pd}$, two cost values $C_{AD}(\mathbf{p}, d)$ and $C_{\mathrm{census}}(\mathbf{p}, d)$ are computed individually.

$C_{AD}(\mathbf{p}, d)$ is defined as the average intensity difference of $\mathbf{p}$ and $\mathbf{pd}$ in RGB channels by Eq. (1). In the original census transform, the brightness value of anchor pixel $\mathbf{p}$ is compared with the brightness values of pixels $N(\mathbf{p})$ in the census window W. In Eq. (2), function $\xi()$ returns 1 if the brightness value $I(\mathbf{p}_n)$ of the neighborhood pixel is higher than the counterpart $I(\mathbf{p})$ of the central pixel and 0 if the brightness value of the neighborhood pixel is lower than that of the central pixel by comparing the brightness values between pixels. Using the concatenation operator $\otimes$, the pixel-based brightness comparison results are encoded into the bit-string $C(\mathbf{p})$. More specifically, $C(\mathbf{p})$, consisting of 0 and 1, refers to the relative brightness distribution of neighborhood pixels on the basis of the central pixel of the window.

By considering only the resultant binary code of the pixel $\mathbf{p}_n$ in the intersection support region $\mathrm{ISR}(\mathbf{p})$, we can reduce the errors caused by irrelevant pixels. This means that $\mathbf{p}_n$ is a pixel in the intersection region of $N(\mathbf{p})$ and $\mathrm{ISR}(\mathbf{p})$. The census transform result is obtained from the hamming distance of the two bit-strings of pixel $\mathbf{p}$ and the corresponding $\mathbf{pd}$

$$C_{AD}(\mathbf{p}, d) = \frac{1}{3}\sum_{i=R,G,B}|I_i^{\mathrm{Left}}(\mathbf{p}) - I_i^{\mathrm{Right}}(\mathbf{pd})|, \tag{1}$$

$$C(\mathbf{p}) = \bigotimes_{\mathbf{p}_n \in \mathrm{ISR}(\mathbf{p}) \cap N(\mathbf{p})} \xi(\mathbf{p}, \mathbf{p}_n),$$
$$\xi(\mathbf{p}, \mathbf{p}_n) = \begin{cases} 1, & \text{if } I(\mathbf{p}) < I(\mathbf{p}_n) \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

The length of the bit-string by census transform depends on the size of the census mask. The proposed method stores the bit-string $C(\mathbf{p})$ by census transform in a one-byte unit. For example, when the census window is 9×7 pixels, 62 bits are generated by the brightness comparison. Then, eight strings of one-byte length are encoded, and the first two bits of the eighth string are "do not care" bits. An exclusive-OR operation of the binary string in the reference image and in the target image is performed in the census transform. A look-up table that contains the exclusive-OR operation results of all two bit-strings (of one-byte length) is used for computation efficiency.

$C_{AD}(\mathbf{p}, d)$ is normalized with the maximum intensity value (255) as Eq. (3). When the AD measure and census transform are combined, Mei et al.[4] used the exponential function. First, it maps different cost measures (AD measure and census transform) to the range [0,1], such that the cost values will not be severely biased by one of the measures. Second, it allows easy control of the influence of the outliers in each cost measure.

A clipping function is used for efficient implementation, instead of the exponential function used in Ref. 4. $C_{N\_AD}(\mathbf{p}, d)$ is aggregated in the intersection support region $\mathrm{ISR}(\mathbf{p}, d)$, which are normalized with the area of the intersection support region area$[\mathrm{ISR}(\mathbf{p}, d)]$ as Eq. (3). The size and shape of the intersection support region is determined for the candidate disparity levels $(0 - d_{\max -1})$. In the same way, $C_{\mathrm{census}}(\mathbf{p}, d)$ is normalized with the area of the intersection of $N(\mathbf{p})$ and $\mathrm{ISR}(\mathbf{p}, d)$, and then clipped in

Eq. (4). Here, λ_{AD} and λ_{census} are the threshold values for the clipping functions. Two parameter values are determined by considering the matching costs of the AD measure and census transform, and $C_{N\_AD}(\mathbf{p}, d)$ and $C_{\text{census}}(\mathbf{p}, d)$ are normalized by λ_{AD} and λ_{census}, respectively. In Eq. (5), the initial matching cost $C(\mathbf{p}, d)$ is computed with $C_{\text{SAD}}(\mathbf{p}, d)$ and $C_{\text{census}}(\mathbf{p}, d)$. To give the relative weight of importance for two cost measures directly, w_{AD} and w_{census} parameters are included

$$C_{N\_AD}(\mathbf{p}, d) = \frac{\min\left[\frac{C_{AD}(\mathbf{p},d)}{255}, \lambda_{AD}\right]}{\lambda_{AD}},$$

$$C_{\text{SAD}}(\mathbf{p}, d) = \frac{\sum_{\mathbf{p}_{\text{n}} \in \text{ISR}(\mathbf{p})} C_{N\_AD}(\mathbf{p}_{\text{n}}, d)}{\text{Area}[\text{ISR}(\mathbf{p}, d)]}, \tag{3}$$

$$C_{\text{census}}(\mathbf{p}, d) = \frac{\min\left\{\frac{\sum \text{Hamming}[C_{\text{left}}(\mathbf{p}), C_{\text{right}}(\mathbf{pd})]}{\text{Area}[\text{ISR}(\mathbf{p},d) \cap N(\mathbf{p})]}, \lambda_{\text{census}}\right\}}{\lambda_{\text{census}}}, \tag{4}$$

$$C(\mathbf{p}, d) = w_{AD} C_{\text{SAD}}(\mathbf{p}, d) + w_{\text{census}} C_{\text{census}}(\mathbf{p}, d). \tag{5}$$

The size and shape of the intersection support region vary greatly with disparity levels. If a census transform is applied to the intersection support regions at every disparity level, its computational load is very high. Hence, a census transform with a fixed window size and shape is applied, and only the resultant binary code of the pixel in the intersection support region is used as Eq. (2). Even if the size of the intersection support region is larger than the census transform window, the result of the census transform with the fixed window is used as $C(\mathbf{p})$.

2.3 *Area Similarity Ratio of Intersection Support Regions*

The proposed method introduces a relative reliability weight based on the similarity ratio of the areas of the cross-based support regions between stereo views. The area of the intersection support region is divided by the area of the cross-based support region in the reference image. The obtained area ratio R_i (0.0 to 1.0) represents the area similarity between the support region in the reference image and target image.

Figure 3 shows the process of computing the area similarity ratio and assigning reliability weights at each disparity level in more detail. The figure shows both the candidate corresponding pixels and their support regions in the target image with respect to the anchor pixel in the reference image. Here, the anchor pixel in the cross-based support region is indicated by a white dot. In the winner takes all (WTA) strategy, the disparity with the minimum cost value is generally selected as the final disparity estimate.[1] Accordingly, the reciprocal value of the reliability weight based on the probability is multiplied by the initial matching cost at each disparity level.

From the Middlebury reference images and their ground truth depth maps, the conditional probability of observing a correct disparity estimate on the condition that the area similarity ratio R_i is given is examined. To compute this probability, the number of cases where correct disparity estimates are obtained from R_i is divided by the total number of image pixels (width × height). The area similarity ratio is divided

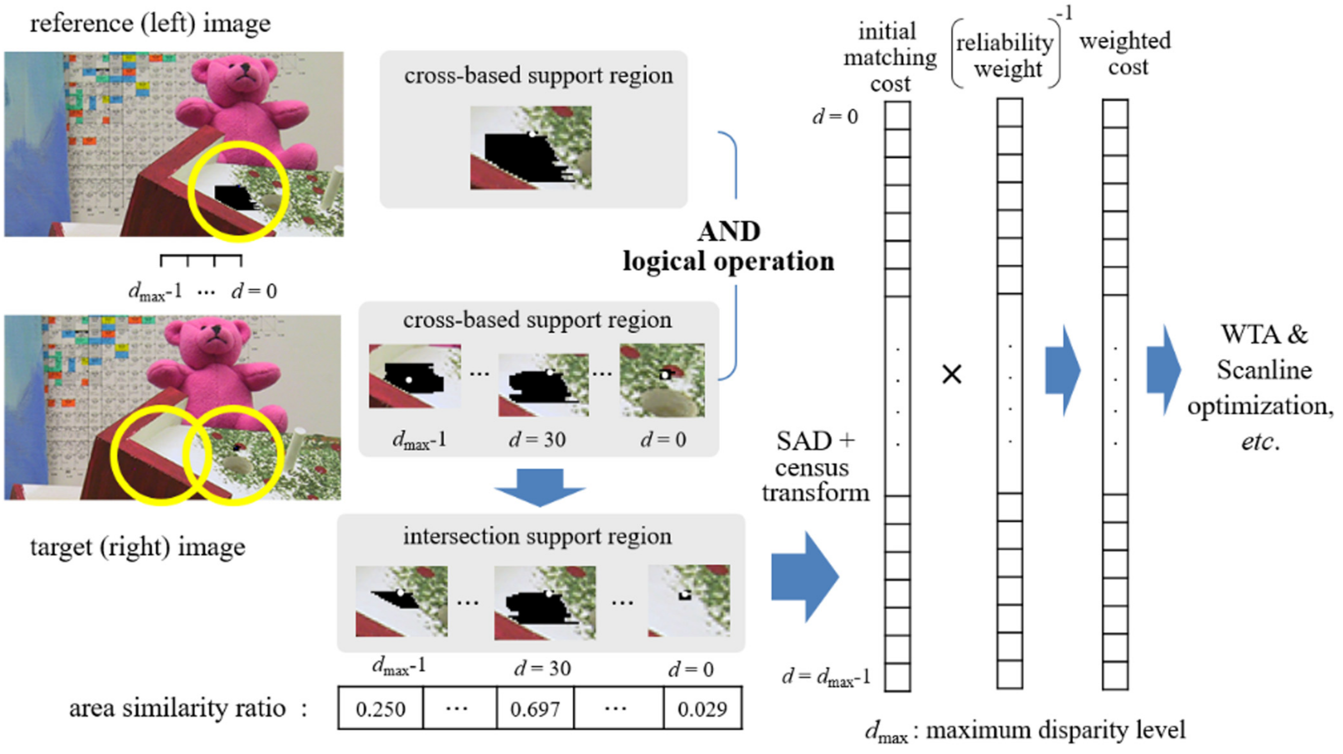


Fig. 3 Process of area similarity ratio computation at each disparity level.

into a constant sampling interval, which covers the range of R_i (8 to 256 levels).

It is common practice to keep separate training and testing datasets. To do so, the probabilities of observing a correct disparity estimation given the area similarity are computed in 2005 to 2006 Middlebury benchmark reference images (14 images). More specifically, 2005 benchmark images (Art, Books, Dolls, Moebius and Reindeer) and 2006 benchmark images (Baby1, Bowling1, Cloth1, Flowerpots, Lampshade1, Midd1, Monopoly, Plastic, and Wood1) are used as training datasets. The 2001 and 2003 Middlebury benchmark reference images (Tsukuba, Venus, teddy, and cones) are used as testing datasets for quantitative performance evaluation. In addition, 2005 and 2006 Middlebury database sets (aloe, laundry, rock1, and cloth4) are used as testing datasets for qualitative performance evaluation.

Figure 4(a) shows the distribution of the averaged conditional probability according to R_i ($i = 16$) in 14 reference images (2005 to 2006 Middlebury benchmark sets). When the cross-based support region in the target image is similar to that in the reference image (i.e., the area similarity ratio is close to 1), a more accurate disparity value is obtained. Here, an extremely small support region (area less than 5×5) is excluded from this estimation procedure because it will yield a small denominator, which may cause ambiguity in the area similarity ratio computation.

By taking the natural log probability, a relative reliability weight according to the area similarity ratio is obtained. In general, since the probability value is too small, its logarithm result may be negative. Here, the probability value is multiplied by 10^5 to make the logarithm results of the minimum probability value positive. Then, the logarithm results are normalized with the value at the last sampling level. Figure 4(b) shows the reciprocal value of the reliability weight according to the R_i. The weighted matching cost distribution of the anchor pixel is examined in the disparity estimation procedure.

2.4 *Disparity Refinement*

Unreliable disparity values could still be obtained owing to occlusions, repetitive structures, and texture-less regions. To eliminate these matching ambiguities, a four-direction scanline optimizer based on a semiglobal matching method is used.[4,13,14] Given the scanline directions (two along the horizontal directions, and two along the vertical directions), the path cost at pixel **p** and disparity d is updated, penalizing the disparity changes between neighboring pixels. The final cost for pixel **p** and disparity d is obtained by averaging the path costs from four directions.

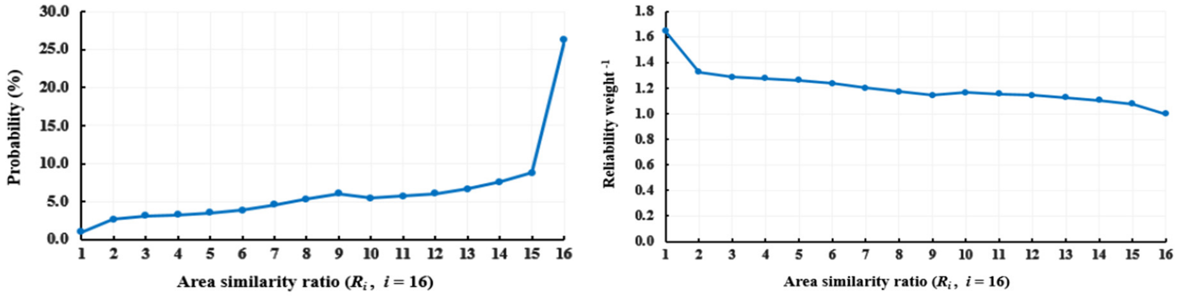
Using left–right consistency (LRC) checking, the matching ambiguity regions can be obtained.[2,3] The edge propagation (EDP) method is then used as an optimization process with color continuity and edge information.[15] The EDP method propagates the disparity values to the peripheral regions, considering the color differences of pixels and the edge costs of regions. When the cost value of an adjacent pixel is small, the disparity values of neighborhood pixels are propagated. The proposed method uses WTA to determine the disparity value in the cost volume, and then performs the LRC check for outlier detection. In addition, unstable disparity estimation values are refined by iterative region voting and proper interpolation procedures.[3]

3 Experimental Results

The experiment was carried out using a PC with Intel(R) Core(TM) i7-6700 CPU @3.40 GHz, NVIDIA Geforce GTX 1070 graphics card, and VS 2013, OpenCV 2.4.13 development environment. The proposed algorithm was implemented on a GPU-based CUDA 8.0 platform with multithreads and parallel programming.

Figure 5 shows the Tsukuba, Venus, teddy, and cones stereo datasets (2001 and 2003), their ground truth disparity maps, and the disparity map results from our method. Table 2 shows four reference benchmark image resolutions and their maximum disparity levels. Table 3 shows quantitative evaluation results by stereo matching algorithms for the Middlebury database set. In Middlebury stereo evaluation, the percentage of bad matched pixels (BMP) is used as the error measure.[1] The BMP is based on counting the disparity estimation errors exceeding a threshold value, and the most commonly used value is 1 pixel. In this paper, the threshold value for the BMP is set to 1. The matching errors of previous methods are listed with their original rank, as reported in the Middlebury benchmark. In this comparison, the area similarity ratio is divided into four sampling intervals (8, 16, 32, and 64 levels).

Table 3 shows that the sampling interval of R_i exhibits some influence on the disparity estimation results. However, when the sampling interval of R_i is too narrow, it becomes difficult for the reliability weight to reflect precisely the correlation between R_i and the correct disparity estimate. When the sampling interval of R_i is 64 levels, the smallest average errors (bold values) are obtained. So, the sampling interval of R_i is set to 64 levels in this experiment. The proposed



(a)

(b)

Fig. 4 (a) Averaged conditional probability according to area similarity ratio and (b) reciprocal value of the reliability weight.

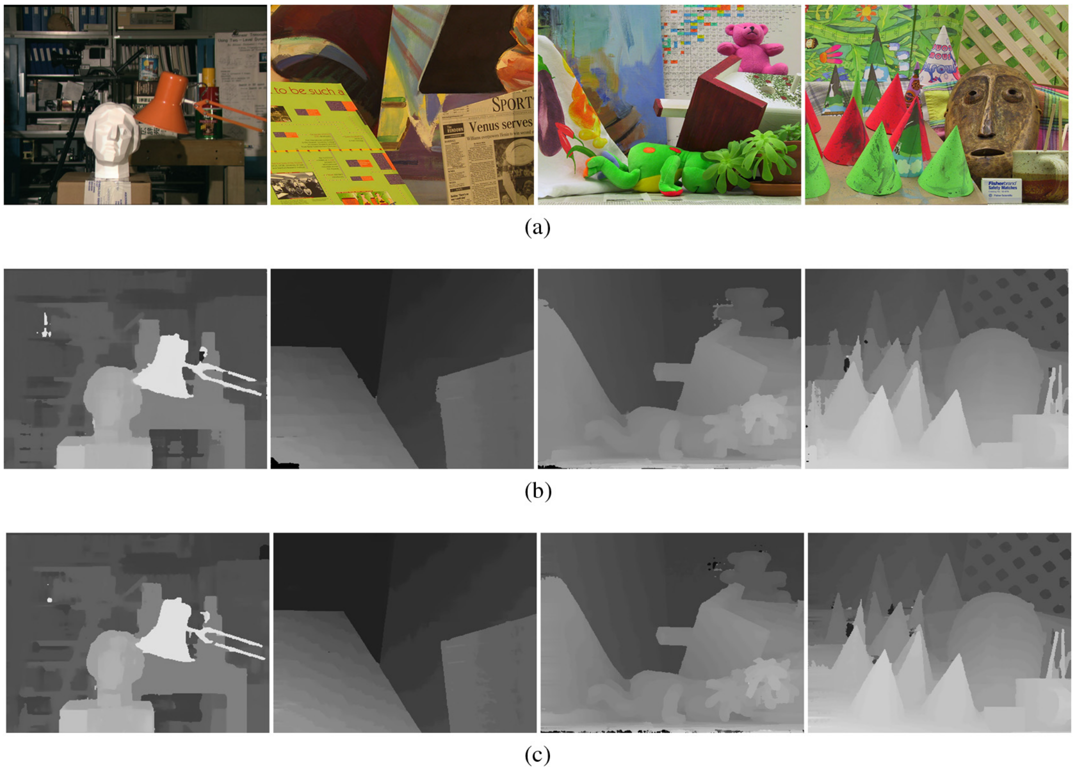


Fig. 5 (a) Tsukuba, Venus, teddy, and cones stereo images (from left to right).[11] Disparity maps (b) by previous method[16] and (c) by the proposed algorithm (R_i, $i = 64$).

Table 2 Four reference stereo images and their parameters.

	Tsukuba	Venus	Teddy	Cones
Resolution	384 × 288	434 × 383	450 × 375	450 × 375
Maximum disparity level	16	20	60	60

method provides better results compared with previous algorithms except the AD-census method.[2,4,7,16,17] Table 3 shows that the performance of the proposed method is better or nearly the same as the AD-census method.[4] Using the adaptive matching window based on the intersection support regions in stereo views, the proposed method can reduce the errors caused by irrelevant pixels in the initial matching cost computation. The experimental results show that the reliability weight based on the area similarity ratio of support regions is helpful for accurate disparity estimation. A previous algorithm based on color segmentation and a plane estimation procedure is also considered. The performance of the method in the Middlebury test is high but involves significant computational load. Here, performances are evaluated in the nonoccluded region, all (including half-occluded) regions, and regions near depth discontinuities, denoted as "non-occ," "all," and "disk," respectively. Here, λ_{AD}, w_{AD}, λ_{census}, and w_{census} are set to 0.1, 0.2, 0.8, and 1.0, respectively. Additionally, the spatial and color control parameters in EDP are set to 40 and 20, respectively.

Reliability measures are used to evaluate how the reliable disparity value is obtained and to reduce the average error of the disparity map.[18,19] Two reliability measures are used to evaluate matching performance: the peak-ratio naive (PKRN) and the maximum likelihood metric (MLM).[19] PKRN computes the ratio of the minimum cost C_1 and the second minimum cost C_2 as Eq. (6). MLM obtains a probability density function for the disparity of the minimum cost. MLM measures the relative minimum cost value compared with the sum of the total minimum cost values as Eq. (7). Here, ε and σ are set to 128 and 8, respectively. More specifically, the confidence methods measure the disparity estimate's likelihood of being correct and generate reliable disparity maps by selecting among multiple hypotheses for each pixel

$$\text{PKRN} = \frac{C_2 + \varepsilon}{C_1 + \varepsilon} - 1, \tag{6}$$

$$\text{MLM} = \frac{e^{-C_1/2\sigma^2}}{\sum e^{-C_1/2\sigma^2}}. \tag{7}$$

Figure 6 shows the confidence maps of the disparity estimation results by the proposed method. EDP in the refinement procedure replaces the matching cost distribution with a new quadric cost distribution based on the stable disparity value. Hence, we examine the disparity estimation results before performing the EDP process. In Fig. 6, the first row shows PKRN confidence maps of the depth estimation, and the second row shows MLM confidence maps. In addition, the first column shows the confidence map of the depth estimation without adaptive window and reliability weight. The second column shows the confidence map of the depth estimation

Table 3 Percentage of BMPs for Middlebury database set.

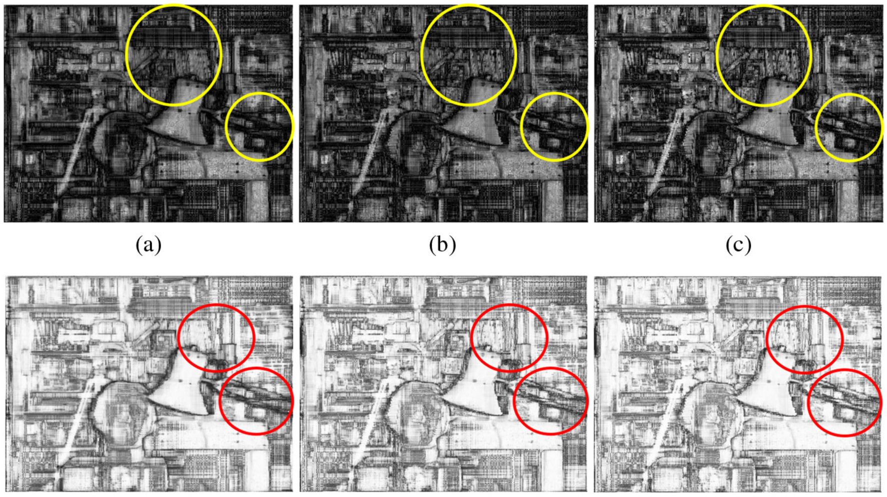
		Tsukuba			Venus			Teddy			Cones			
		Non-occ	All	Disc	Non-occ	All	Disc	Non-occ	All	Disc	Non-occ	All	Disc	Aver.
AD-census[4]		1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.90	2.42	7.25	6.95	3.97
Proposed method	8	1.82	2.33	6.91	0.12	0.31	1.47	3.59	6.95	10.19	1.80	7.36	5.34	4.01
	16	1.34	1.87	6.40	0.15	0.35	1.85	3.69	6.93	10.48	1.76	7.31	5.21	3.95
	32	1.30	1.84	6.35	0.15	0.35	1.89	3.72	7.00	10.51	1.77	7.25	5.22	3.95
	64	**1.29**	**1.81**	**6.42**	**0.16**	**0.35**	**1.95**	**3.73**	**6.97**	**10.53**	**1.76**	**7.17**	**5.19**	**3.94**
DoubleBP[7]		0.88	1.29	4.76	0.13	0.45	1.87	3.53	8.30	9.63	2.90	8.78	7.79	4.19
Previous method[16]		1.71	2.46	7.54	0.15	0.51	1.73	4.43	10.3	12.70	2.80	8.81	7.88	5.09
AdaptWeight[2]		1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67
DCB grid[17]		5.90	7.26	21.00	1.35	1.91	11.20	10.50	17.20	22.20	5.34	11.90	14.90	10.90

with adaptive window and no reliability weight consideration. The third column shows the confidence map of the depth estimation with both adaptive window and reliability weight. Confidence maps using PKRN and MLM are nonlinearly scaled for visualization. Here, brighter regions with high confidence values represent more reliable disparity estimates. Figure 6 shows that the proposed method with adaptive window and reliability weight can improve disparity estimation performance in the detailed parts, such as the book shelf and light stand (marked in colored circles).

Table 4 shows the modular computation performance on CUDA implementation for stereo images. In many stereo matching studies, the computation loads have been evaluated based on the fixed maximum disparity levels of benchmark stereo images.[3,20] In Table 2, the resolution and the maximum disparity level of teddy images are 450 × 375 and 60, respectively, which are the same as those of Cones images. The resolution and maximum disparity level of both teddy and cones images are higher than the other two images. Overall, computation loads are highly dependent on both the image resolution and maximum disparity level. This means that less matching cost computations are performed in Tsukuba and Venus images.

Initially, cross-based support regions in stereo views and their intersection regions along the disparity level are built. The area similarity ratio of the intersection region and the census transform within the 9×7 window are also computed at this stage. In the initial matching cost computation step, two procedures have almost the same processing time: (1) finding the pixels in the intersection region and computing AD estimates, and (2) combining AD aggregation results with census transform results and considering the reliability weight in the matching cost computation. The proposed method obtains the final disparity map in 20.12 to 139.17 ms (7.2 to 49.7 frame/s).

In this experiment, the average sizes of support regions in cones and teddy images are 235.01 and 454.77 pixels,


(a) (b) (c)

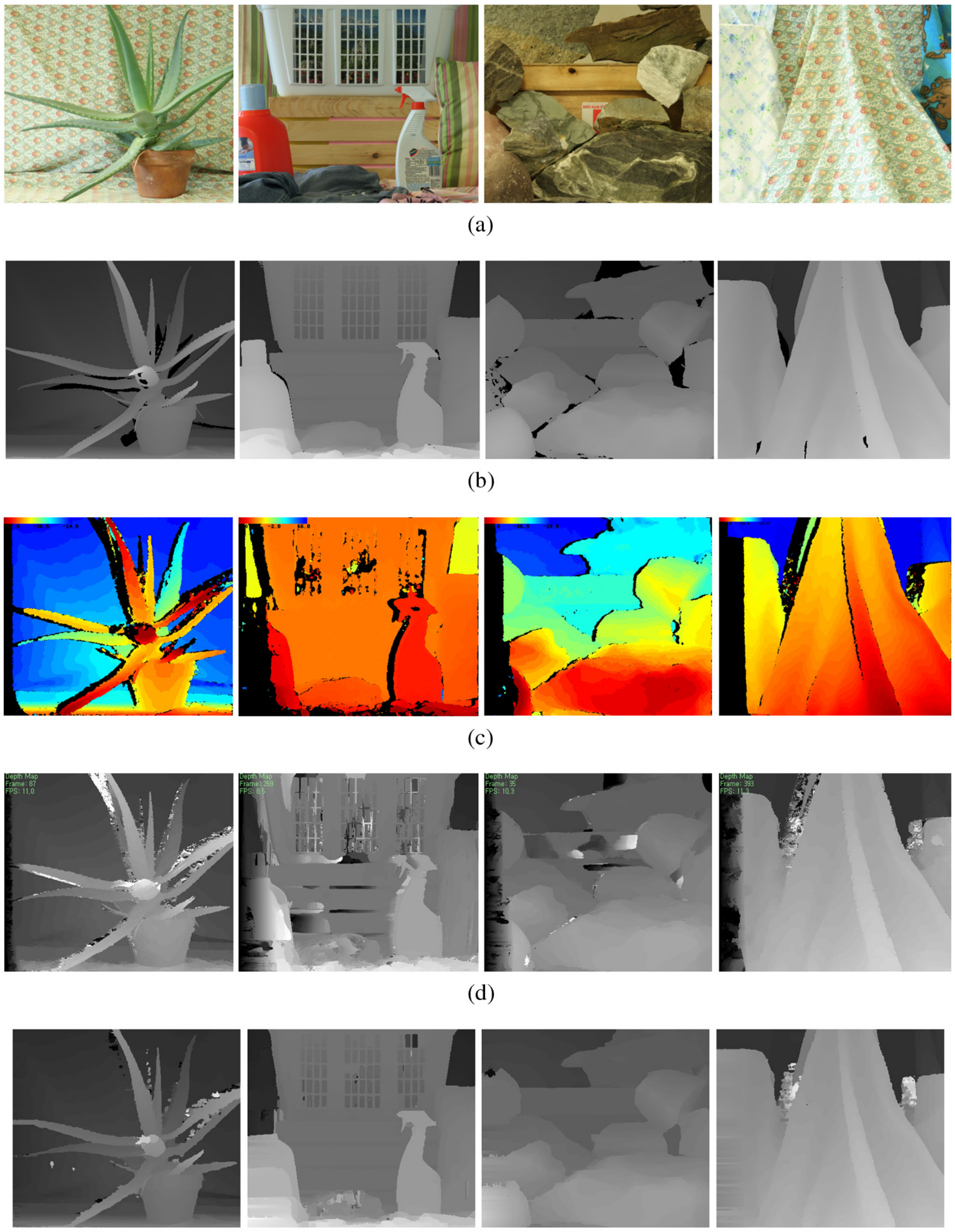
(d) (e) (f)

Fig. 6 PKRN (first row) and MLM (second row) confidence maps: (a), (d) without adaptive window and reliability weight; (b), (e) with adaptive window and no reliability weight; and (c), (f) with both adaptive window and reliability weight.

Table 4 Computation time (*ms*) of modules.

	Initial setting	Matching cost computation	Scanline optimization	LRC check and EDP	Refinement	Total
Tsukuba	4.08	5.34	7.32	2.24	1.14	20.12
Venus	6.95	9.14	14.34	3.42	1.37	35.22
Teddy	15.03	22.27	91.72	7.98	2.17	139.17
Cones	14.59	22.25	91.71	8.02	2.57	139.26

respectively. This means that there are more homogeneous color regions in teddy images than in cones images. More pixels are examined to construct the support regions in teddy images. Therefore, in teddy images, the computation time of the initial step with the support region construction is longer than that in Cones images. The size of the support region has little effect on the matching cost computation step, because the integral image of matching cost is



(a)

(b)

(c)

(d)

(e)

Fig. 7 (a) Stereo images (left view) and (b) ground truth.[11] Disparity maps by (c) MGM method,[22] (d) DCB grid,[17] and (e) proposed method (R_i, $i = 64$).

employed in the cost aggregation. The numbers of occlusion pixels by LRC check in teddy images and cones images are 18,404 and 19,479, respectively. Then, the refinement step (LRC check and iterative region voting) is performed to overcome occlusion regions. In Table 4, the computation of the refinement step in cones images takes longer time than that in teddy images. Table 4 shows that the scanline optimization step requires longer computation time. To improve performance further, we will consider another optimization method based on parallel GPU architecture for computation efficiency.[21]

In Figs. 7 and 8, the proposed method is qualitatively compared with previous methods.[17,22] The test is performed for 2005 and 2006 Middlebury database sets (aloe, laundry, rock1, and cloth4) and the book arrival sequence,[23] a real-world stereo video clip captured in an uncontrolled environment. Figures 7(c)–7(e) show the disparity maps by the MGM method, DCB grid method, and proposed method, respectively. In the MGM method, a refinement procedure is not included and the disparity results are expressed in pseudo color. However, we can see roughly matching performance by the MGM method. The disparity maps by the proposed method have more distinct boundaries of objects and are less noisy. However, further considerations to eliminate errors in the occluded and textureless regions (such as backgrounds in rock1 image) are needed. The reference images (the 1st and 20th frames) and their disparity results are shown in Fig. 8, respectively. The next two columns show the results produced by previous methods. (DCB grid's source code is downloaded from author's project website at https://www.cl.cam.ac.uk/research/rainbow/projects/dcbgrid/, and disparity results by the MGM method are obtained using an online demo on author's website at http://dev.ipol.im/~faccilol/mgm.) The last column shows the results by the proposed method. In the same manner, the proposed method is compared with two algorithms on outdoor scene image sequence, as shown in Fig. 9.

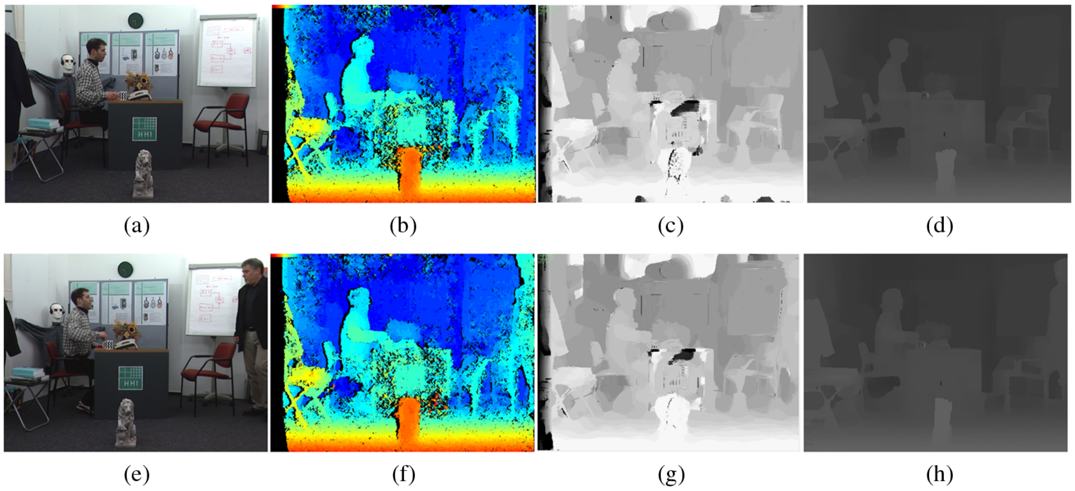


Fig. 8 Snapshots of book arrival stereo sequence from FhG-HHI database:[19] (a), (e) Reference frame. Disparity maps by (b), (f) MGM method;[22] (c), (g) DCB grid;[17] and (d), (h) proposed method (R_i, $i = 64$).

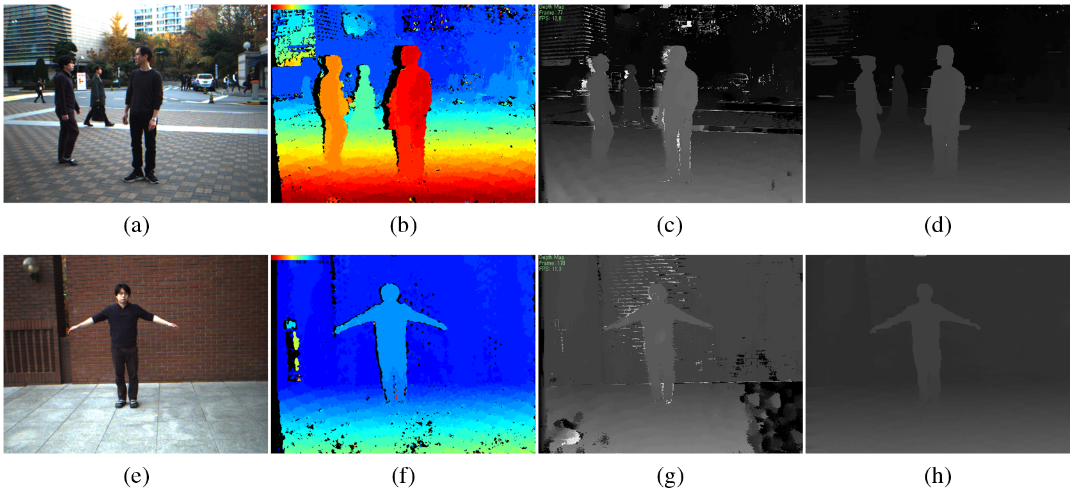


Fig. 9 (a), (e) Outdoor scene images. Disparity maps by (b), (f) MGM method;[22] (c), (g) DCB grid;[17] and (d), (h) proposed method (R_i, $i = 64$).

4 Conclusion

This paper presents a near-real-time stereo matching method using cross-based support regions in both the reference and target images. The proposed method obtains the intersection region of the cross-based support regions in stereo views. The intersection region is used as an adaptive matching window for initial matching cost computation. When the area of the support region in the reference image is similar to that in the target image, the candidate disparity is likely to be the correct disparity value. The proposed method computes the reliability weight based on this probability from the Middlebury reference database sets. Experimental results show that the proposed method with CUDA implementation provides improved matching accuracy and processing efficiency.

Acknowledgments

This research was supported by the Chung-Ang University Research Scholarship Grants in 2016, and the MIST (Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information and Communications Technology Promotion) (20170001000011001).

References

1. D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vision* **47**(1), 7–42 (2002).
2. K. Yoon and I. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 650–656 (2006).
3. K. Zhang, J. Lu, and G. Lafruit, “Cross-based local stereo matching using orthogonal integral images,” *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7), 1073–1079 (2009).
4. X. Mei et al., “On building an accurate stereo matching system on graphics hardware,” in *IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pp. 467–474 (2011).
5. G. Van Meerbergen et al., “A hierarchical symmetric stereo algorithm using dynamic programming,” *Int. J. Comput. Vision* **47**(1), 275–285 (2002).
6. L. Wang et al., “High quality real-time stereo using adaptive cost aggregation and dynamic programming,” in *Proc. of Int. Symp. on 3D Data Processing, Visualization and Transmission*, pp. 798–805 (2006).
7. Q. Yang et al., “Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling,” *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(3), 492–504 (2009).
8. Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001).
9. S. B. Kang, R. Szeliski, and J. Chai, “Handling occlusions in dense multi-view stereo,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 103–110 (2001).
10. O. Veksler, “Fast variable window for stereo correspondence using integral images,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 556–561 (2003).
11. D. Scharstein, Middlebury stereo vision,” http://vision.middlebury.edu/stereo/ (16 December 2015).
12. S. Zhu and Z. Li, “Local stereo matching using combined matching cost and adaptive cost aggregation,” *KSII Trans. Internet Inf. Syst.* **9**(1), 224–241 (2015).
13. H. Hirschmüller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(9), 1582–1599 (2009).
14. H. Hirschmüller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 328–341 (2008).
15. X. Sun et al., “Real-time local stereo via edge-aware disparity propagation,” *Pattern Recognit. Lett.* **49**, 201–206 (2014).
16. S. Kang and H. Hong, “Near-real-time stereo matching method using temporal and spatial propagation of reliable disparity,” *Opt. Eng.* **53**(6), 063107 (2014).
17. C. Richardt et al., “Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid,” *Lect. Notes Comput. Sci.* **6313**, 510–523 (2010).
18. D. Pfeiffer, S. Gehrig, and N. Schneider, “Exploiting the power of stereo confidences,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 297–304 (2013).
19. X. Hu and P. Mordohai, “Evaluation of stereo confidence indoors and outdoors,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1466–1473 (2010).
20. N. Ma et al., “Segmentation-based stereo matching using combinatorial similarity measurement and adaptive support region,” *Optik* **137**, 124–134 (2017)
21. D. Hernandez-Juarez et al., “Embedded real-time stereo estimation via semi-global matching on the GPU,” *Procedia Comput. Sci.* **80**, 143–153 (2016)
22. G. Facciolo, C. Franchis, and E. Meinhardt, “MGM: a significant more global matching for stereovision,” in *Proc. of British Machine Vision Conf.*, pp. 90.1–90.12 (2015).
23. “Mobile 3DTV content delivery optimization over DVB-H system, The book arrival data sets,” http://sp.cs.tut.fi/mobile3dtv/stereo-video/ (16 March 2011).

SangYoon Lee received his BS degree in electronic engineering from Ho-Seo University in 2014. He received his MS degree from the Department of Imaging Science and Arts, GSAIM, Chung-Ang University, in 2016. Currently, he is pursuing a PhD degree in the School of Integrative Engineering, Chun-Ang University. His research interests include computer vision and augmented reality.

Hyunki Hong received his BS, MS, and PhD degrees in electronic engineering from Chung-Ang University, Seoul, Korea, in 1993, 1995, and 1998, respectively. Since 2000, he has been a professor in the School of Integrative Engineering, Chung-Ang University. From 2002 to 2003, he was a postdoctoral researcher in the Department of Computer Science and Engineering at the University of Colorado, Denver. His research interests include computer vision and augmented reality.