



Research article

A new stereo matching algorithm based on improved four-moded census transform and adaptive cross pyramid model

Zhongsheng Li¹, Jianchao Huang^{2,*}, Wencheng Wang¹ and Yucai Huang¹

¹ School of Traffic Engineering, Fujian Polytechnic of Water Conservancy and Electric Power, Yongan 366000, China

² School of Electric Power Engineering, Fujian Polytechnic of Water Conservancy and Electric Power, Yongan 366000, China

* **Correspondence:** Email: 1422304254@qq.com.

Abstract: Stereo matching is still very challenging in terms of depth discontinuity, occlusions, weak texture regions, and noise resistance. To address the problems of poor noise immunity of local stereo matching and low matching accuracy in weak texture regions, a stereo matching algorithm (iFCTACP) based on improved four-moded census transform (iFCT) and a novel adaptive cross pyramid (ACP) structure were proposed. The algorithm combines the improved four-moded census transform matching cost with traditional measurement methods, which allows better anti-interference performance. The cost aggregation is performed on the adaptive cross pyramid structure, a unique structure that improves the traditional single mode of the cross. This structure not only enables regions with similar color and depth to be connected but also achieves cost smoothing across regions, significantly reducing the possibility of mismatch due to inadequate corresponding matching information and providing stronger robustness to weak texture regions. Experimental results show that the iFCTACP algorithm can effectively suppress noise interference, especially in illumination and exposure. Furthermore, it can markedly improve the error matching rate in weak texture regions with better generalization. Compared with some typical algorithms, the iFCTACP algorithm exhibits better performance whose average mismatching rate is only 3.33%.

Keywords: stereo matching; improved FCT; cross pyramid; adaptive window

1. Introduction

The topic of stereo matching has witnessed an upsurge in popularity in the field of computer vision, and has been widely applied in 3D reconstruction [1], 3D object detection [2], autonomous driving [3], and so on. The process of stereo matching includes obtaining disparity information from

the left and right images taken by the binocular stereo camera and then deducing depth information, etc. Currently, there are a number of stereo matching algorithms. According to the differences in constraint range, these algorithms can be divided into two categories: local stereo matching algorithms and global stereo matching algorithms. Local stereo matching algorithms estimate the disparity based on the local information in the pixel window and compute the local optimal solutions to the disparity. They are characterized by good real-time performance. The commonly used local stereo matching algorithms include Variable Support Window (VSW), Adaptive Support Weight (ASW), and multi-window algorithm [4]. Different from the local stereo matching, global stereo matching algorithms compute the disparity value by finding the minimum value of the constructed global energy function with a high degree of matching accuracy but are computationally complex. Common global stereo matching algorithms encompass belief propagation (BP) [5, 6], dynamic programming (DP) [7], and graph cut (GC) algorithms [8]. Scharstein and Szeliski [9] summarized the previous stereo matching algorithms and proposed that their process be divided into four steps: matching cost computation, cost aggregation, disparity computation, and disparity refinement. Among them, cost computation and cost aggregation contribute to the accuracy of matching.

The matching cost is to measure the correlation between the pixels to be matched and the candidate pixels. Kok and Rajendran [10] made detailed comments on frequently-used measures: sum of squares difference (SSD), sum of absolute difference (SAD), normalized cross-correlation (NCC), and rank transform (RT). By means of these methods, we can presume that each left-image pixel and its corresponding right-image pixel have the same intensity values. Therefore, they cannot be utilized for outdoor images with remarkable differences in lighting intensity. Hirschmuller [11] used the probability distribution function as a matching measure to compute the matching cost. But the disadvantages are obvious due to the complex principle, the computing inefficiency, and the poor anti-interference performance. Stein [12] showed that using census transform (CT) to compute the matching cost can effectively improve the anti-interference performance in the cost volume, but the matching effect in the weak texture region is not satisfactory. Mei et al. [13] proposed a method combining the SAD and CT. This method overcomes the poor matching effect of the CT in the weak texture regions because it combines the advantages of the SAD and the CT and guarantees high matching accuracy.

The fundamental purpose of cost aggregation is to ensure the cost value to accurately reflect the correlation between pixels. The aggregation cost of local stereo matching is deeply affected by the aggregation window. The large window easily leads to error matching in the regions with discontinuous depth, while the small one results in error matching in the weak texture regions. To solve those problems, Zhang et al. [14] proposed forming a local cross adaptive window for cost aggregation through double thresholds of color and distance, which is less effective in matching in weak texture regions but is less time-consuming. Although the window of the filter window proposed by Hosni et al. [15] is fixed, its advantage is that the computational complexity is independent of the window size. By analyzing those cost aggregation methods, it can be found that they only perform cost aggregation on a single scale layer, without considering that the feature information of the obtained object is a coarse-to-fine process, which results in a higher error matching rate in the weak texture region as a whole. Therefore, Zhang et al. [16] simulated the human visual system and proposed to build a cross-scale cost aggregation (CSCA) framework to incorporate the cost volume of multiple scales to obtain the target cost volume. Although this framework contributes to a good

disparity map, the demerits of the high error matching rate in weak texture regions and the time-consuming process still remain. Later, Pang et al. [17] proposed an adaptive multi-scale cost body construction and aggregation stereo matching algorithm, which has strong adaptability to different texture regions and good robustness, but the matching accuracy of this algorithm in the parallax discontinuity region needs to be improved.

Although research on stereo matching algorithms have made great progress in recent years [18–21], matching at the correct disparity is challenging in real life due to the photometric and geometric distortions introduced by the change of viewpoint and by ambiguities caused by low texture or repetitive patterns in the scene. To solve the above problems, a novel algorithm, iFCTACP, which multiple matching cost measurement methods and an adaptive cross pyramid structure is proposed. The algorithm first implements the fusion of truncated color absolute difference, truncated gradient absolute difference, and the iFCT (an improved four-moded census transform proposed in this paper). Second, in ACP (an adaptive cross pyramid structure proposed in this paper), gradient information is used to distinguish between weak texture regions and rich texture regions, and a new adaptive window is constructed by using a variable color threshold according to the region of pixels.

In summary, the contributions of this paper are threefold:

- 1) A novel cost metric, iFCT, is presented, which is robust to noise such as illumination and exposure. It can not only improve the matching accuracy but also effectively solve the problem of center pixel distortion. It is a supplement and amendment to the traditional cost metric methods.
- 2) A multi-layer adaptive cross pyramid (ACP) structure is proposed for cost aggregation. The ACP can realize the cross-regional connection and information transmission, significantly reducing the possibility of mismatch due to lack of corresponding matching information. This is an improvement on the shortcomings of traditional cross, especially in weak texture regions.
- 3) Qualitative and quantitative experiments demonstrate the effectiveness of the algorithm, iFCTACP. The optimal parameters are also explored for different situations.

2. iFCTACP algorithm description

The input of the proposed algorithm is an epipolar corrected image. The process consists of four steps: 1) Downsample the input image to obtain multi-scale image, and compute the cost of each scale image by the absolute difference of intensity (AD), iFCT, and gradient fusion, to generate the pyramid initial cost volume. 2) Carry out adaptive cross cost aggregation based on the initial pyramid cost volume. Then, apply the regularization term to optimize the consistency problem of adjacent scales of the pyramid. 3) Conduct scanning linear optimization to further improve the accuracy of the cost aggregation and reduce the matching errors. Then, the initial disparity value of each pixel point is computed using winner-takes-all (WTA) strategy. 4) Optimize the unreliable disparity by disparity refinement (left-right consistency check, disparity filling, weighted median filtering). Finally, output the disparity map.

The overall process of the iFCTACP algorithm is shown in Figure 1.

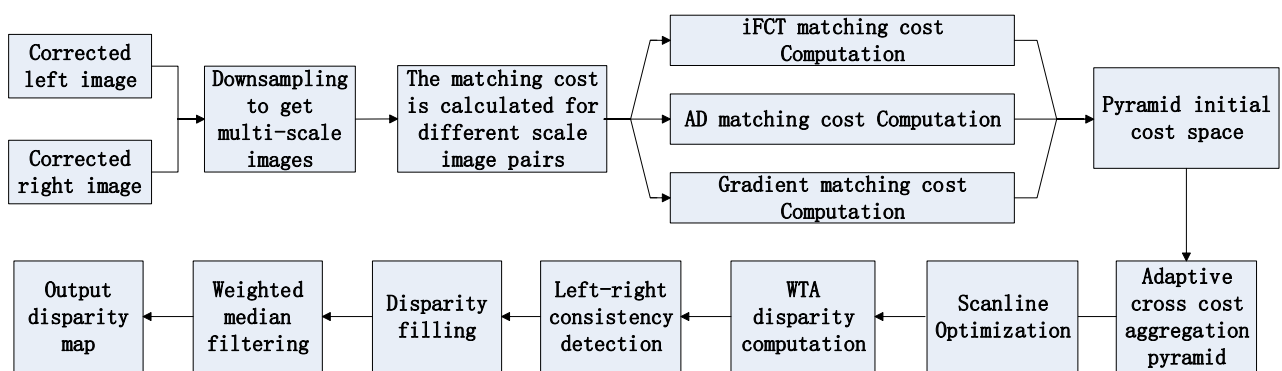


Figure 1. Overall process of the iFCTACP algorithm.

2.1. Cost computation

The purpose of matching cost computation is to measure the correlation between the pixel in the left image and the pixel to be matched in the right image under different disparities. The CT matching cost has been widely used due to its robustness to weak texture regions. It constructs the description code of the center pixel point by comparing the gray value of the center pixel and its neighborhood. The Hamming distance between the codes is used as the matching cost, as defined in Eq (2.1).

$$\begin{cases} \xi(p, q) = \begin{cases} 1, I_p < I_q \\ 0, else \end{cases} \\ B(p) = \bigotimes_{q \in N_p} \xi(p, q) \\ C_{cen}(p, d) = Hamming[B_L(p), B_R(p_d)] \end{cases}, \quad (2.1)$$

where p is a pixel in the left image whose coordinates are (x, y) ; q is a pixel in the neighborhood of p , and I_p and I_q are the gray values of p and q ; p_d represents the corresponding pixel of p with disparity d in the right image, and the coordinates are $(x - d, y)$; N_p is the neighborhood of p ; \bigotimes indicates bitwise connection; $B(p)$ represents the census code of p ; $C_{cen}(p, d)$ denotes the Hamming distance between the codes of pixels, p and p_d .

The definition above reflects that CT transform is highly dependent on the gray value of the central pixel and cannot completely describe the information within the rectangular transform window. In real life, it is inevitable that the image captured by camera will be affected by noise, which will definitely affect the accuracy of the final matching result. To address this problem, Ma et al. [22] proposed the MCT, which replaces the central pixel value with the mean value of the pixels in the window. To a certain extent, the noise robustness of CT is improved. Nevertheless, the mean value is susceptible to the maximum and minimum value of pixels in the window. MRCT, presented by Lai et al. [23], replaces the original gray value of the neighborhood pixel whose relative distance is greater than one unit from the central pixel with the values interpolation of the four pixels surrounding the neighborhood pixel. Although the error matching of MRCT is reduced to some degree, the anti-interference ability is still poor. Lee et al. [24] proposed the SCT, which compares the luminance of pixels at a certain distance along the symmetric pattern within the matching window. The anti-interference ability has been improved, but the algorithm is more complex.

In this paper, we propose iFCT as shown in Algorithm 1 with pseudo-code. It not only improves the matching accuracy, but also effectively solves the central pixel distortion problem. These two improvements benefit from taking the mean pixel value $\bar{I}(p_x, p_y)$ (Algorithm 1, Lines 5~12) within the $m \times n$ window of the center pixel p as another constraint, and using the intensity of the matched pixels within the window to compare with the center pixel value $I(p_x, p_y)$ and the average value of its neighborhood $\bar{I}(p_x, p_y)$ (Algorithm 1, Lines 13~28), as shown in Figure 2. In this way, a new census transform is built. We assign the value of $\bar{I}(p_x, p_y)$ as:

$$\bar{I}(p_x, p_y) = \frac{\sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} I(p_x + i, p_y + j)}{m \times n}, \quad (2.2)$$

where $I(p_x, p_y)$ is the center pixel of the window, and $I(p_x + i, p_y + j)$ is the intensity of each pixel in the window. The $I(p_x + i, p_y + j)$ is compared with $I(p_x, p_y)$ and $\bar{I}(p_x, p_y)$. The new census transform is expressed as:

$$T(p_x, p_y) = \bigotimes_{i \in m} \bigotimes_{j \in n} \xi(I(p_x, p_y), I(p_x + i, p_y + j), \bar{I}(p_x, p_y)), \quad (2.3)$$

where the operator \otimes is a bit concatenation operation, and the auxiliary function ξ , which has four modes, is defined as:

$$\xi(a, b, c) = \begin{cases} 01, & \text{if } a < b < c; \\ 10, & \text{if } a > b > c; \\ 00, & \text{if } b \leq \min(a, c); \\ 11, & \text{if } b \geq \max(a, c). \end{cases} \quad (2.4)$$

We call the improved four-mode census transform as iFCT.

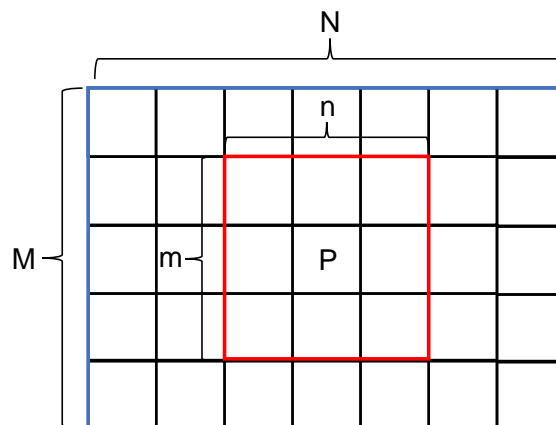


Figure 2. Diagram of IFCT window.

Algorithm 1: Pseudocode for iFCT method

```

Data: The initial image I;
Result: Bit string, census(x,y);
1 for y = M to I.height - M do
2   for x = N to I.width - N do
3     sum = 0, count = 0;
4     for u = -m/2 to m/2 do
5       for v = -n/2 to n/2 do
6         sum+ = I(px + u, py + v);
7         count++;
8       end
9     end
10     $\bar{I}(p_x, p_y) = \text{sum}/\text{count};$ 
11    for i = -M/2 to M/2 do
12      for j = -N/2 to N/2 do
13        gray = I(px + i, py + j);
14        census_tem << 1;
15        if gray < I(px, py) && gray >  $\bar{I}(p_x, p_y)$ , then
16          census_tem << 1, census_tem+ = 1;
17        else if gray > I(px, py) && gray <  $\bar{I}(p_x, p_y)$ , then
18          census_tem+ = 1, census_tem << 1, census_tem+ = 1;
19        end
20        else if gray ≤ min(I(px, py),  $\bar{I}(p_x, p_y)$ ) then
21          census_tem << 1;
22        end
23        else if gray ≥ max(I(px, py),  $\bar{I}(p_x, p_y)$ ) then
24          census_tem+ = 1, census_tem << 1, census_tem+ = 1;
25        end
26      end
27    end
28    census(x,y) = census_tem;
29  end
30 end

```

Figure 3 shows the calculation process of iFCT and CT with a neighborhood window of 5×5 , where m, n is taken as 3. When the center pixel is affected by noise and changed from 72 to 120, the conventional CT transform code changes significantly, while the iFCT transform code remains unchanged thereby, the anti-interference performance is improved.

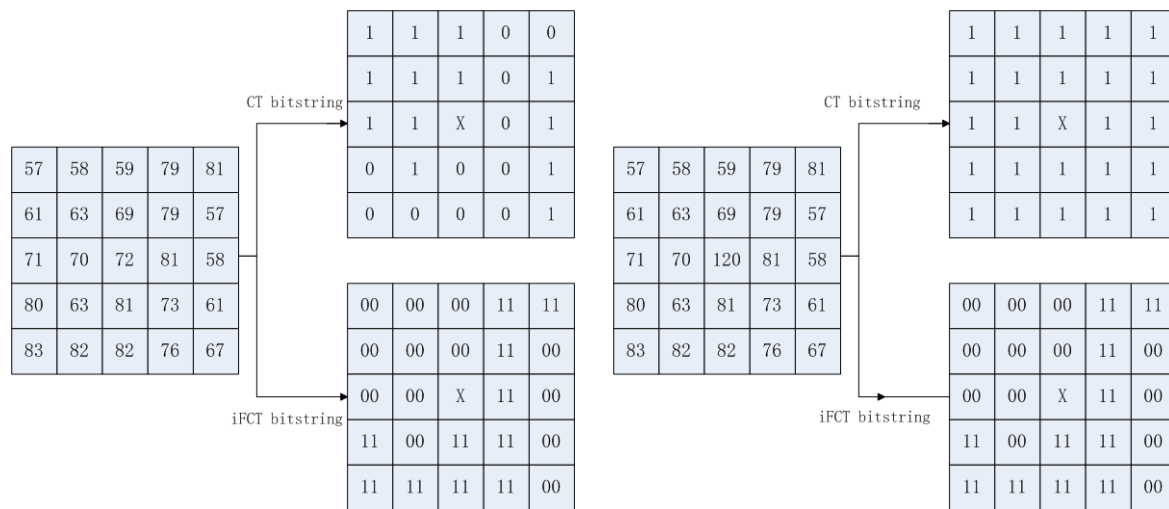


Figure 3. The calculation process of iFCT and CT.

In spite of the simplicity and speed of the cost computation of AD, its sensitivity to noise and many error matchings in the obtained disparity map are the lingering problems. In contrast, gradient and CT have better robustness to noise. To improve the matching accuracy and the robustness under noisy conditions, this paper integrates iFCT with the method proposed by Hosni et al. [25], which can be expressed as the following equation:

$$C'(i, d) = (1 - \alpha) \min(\|I(i) - I'(i_d)\|, \tau_{AD}) + \alpha \cdot \min(\|\nabla_x I(i) - \nabla_x I'(i_d)\|, \tau_{Grad}) \quad (2.5)$$

where i is the pixel in left image, i_d is the matching pixel of i in right image, d is the disparity between i and i_d ; $I(i)$ and $I'(i_d)$ denote the color vectors of i and i_d ; τ_{AD} denotes the color intensity truncation value, τ_{Grad} is the gradient truncation value; α denotes the balance factor between intensity and gradient; and $C'(i, d)$ is the matching cost volume formed by the combination of the absolute differences of truncated color and truncated gradient. Inspired by the idea of normalized fusion [13], this paper uses a variant fusion equation integrates $C'(i, d)$ to compute the initial matching cost value, as follows:

$$C(i, d) = (1 - e^{-\frac{C_{cen}}{\lambda_{cen}}}) + (1 - e^{-\frac{C'}{\lambda'}}), \quad (2.6)$$

where λ_{cen} and λ' are the weights composed of different costs, which can be adjusted to get better matching effect, and $C(i, d)$ is the final matching cost volume.

Figure 4 shows two final disparity maps with the Middlebury teddy data set [26]. One is the results of the cost computation combined with the absolute differences of truncated color and truncated gradient proposed by Hosni et al. [25]. The other is the result of the cost computation fused with iFCT. In order to make the results more convincing, we do not do any other post-processing. The comparisons show that the latter produces better disparity maps than those of the former in the repeating texture regions (marked with red circle) and weak texture regions (marked with yellow ellipse).

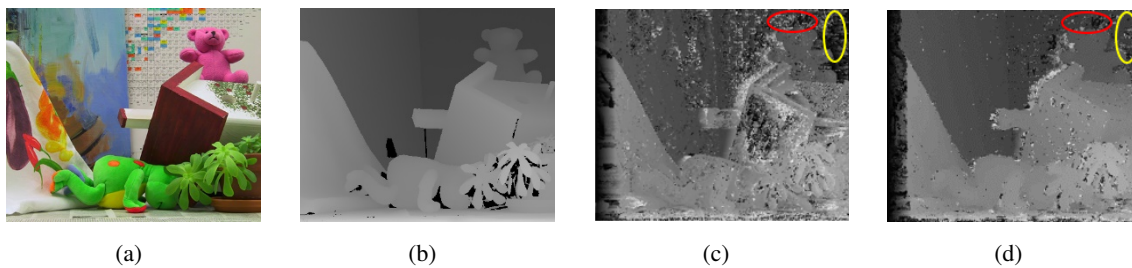


Figure 4. Teddy's initial disparity map based on different matching costs. (a) Original left image. (b) Ground truth disparity map. (c) The result combined with the absolute differences of truncated color and truncated gradient. (d) The result fused with iFCT.

2.2. Cost aggregation of adaptive cross pyramid structure

Currently, most matching cost aggregation methods process the original resolution of the image directly. They can obtain high-quality disparity values in rich texture regions, but in weak texture regions they are prone to mismatch. In order to combine multi-scale information and make full use of coarse-scale information, Zhang et al. [16] proposed a cross-scale model to aggregate the multi-scale matching cost by downsampling to obtain multi-scale information. This model added a generalized L2 regularization term to the optimization goal to make the matching cost consistent on adjacent scales.

One of the essential difficulties in building crossed-based support windows is how to define appropriate rules to include pixels with similar disparity values in the same window. Zhang et al. [14] constructed cross-based support windows by using color similarity and a constant color similarity threshold. However, it cannot handle both discontinuous depth regions and weak texture regions simultaneously. To tackle this problem, Mei et al. [13] analyzed the shortcomings of the cost aggregation method of the cross window proposed by Zhang et al. [16] and improved the matching accuracy in the disparity discontinuous region by setting stricter constraints to determine the arm length of the cross window. However, its fixed threshold setting has a deep impact on the matching accuracy. These methods do not adjust the color and spatial thresholds depending on the type of area where the pixel is located, resulting in an over-extended window arm and more false disparity information in weak texture areas and edge areas.

Therefore, we present a multi-layer adaptive cross pyramid (ACP) structure for cost aggregation. The proposed cost aggregation method is summarized in Algorithm 2. The arm length of the cross window is determined by gradient information and variable color threshold, and its structure is shown in Figure 5. At first, the gradient threshold is set to classify whether the pixel belongs to the weak texture region or the edge region. For weak texture regions, larger color and distance thresholds are set to obtain more correct disparity information. For the edge regions, the color and distance thresholds are reduced to prevent obtaining too much incorrect disparity information. The improved window arm length constraint for the weak texture regions can be formulated as:

$$\begin{cases} G(p) < \beta \\ D_s(p_l, p) < L_1 \\ D_c(p_l, p) < \tau_1, D_c(p_l, p_l + (1, 0)) < \tau_1 \\ D_c(p_l, p) < \tau_2, \text{if } L_2 < D_s(p_l, p) < L_1 \end{cases}, \quad (2.7)$$

where, β is the gradient threshold to determine whether the pixel belongs to the weak texture region

or the edge region; $D_s(p_l, p)$ is the spatial distance between pixel p_l and pixel p ; $D_c(p_l, p)$ is the RGB color difference between pixel p_l and pixel p ; τ_1 and τ_2 are color thresholds and $\tau_2 < \tau_1$; L_1 and L_2 are distance thresholds and $L_2 < L_1$; and τ_2 is a variable color threshold. When $L_2 < D_s(p_l, p) < L_1$, the τ_2 is strictly limited to prevent including more pixels with different depths. It changes nonlinearly with the current color and arm length. The longer the arm length is, the more strictly the color threshold τ_2 is set. It is defined as follows:

$$\tau_2 = -e^{\left(-\frac{(L_1 - D_s(p_l, p))}{\lambda_L}\right)} \cdot \tau_2 + \tau_2, \quad (2.8)$$

where λ_L is the adjustment parameter that regulates the effect of $D_s(p_l, p)$ on the final τ_2 . For the edge and discontinuous regions, it is necessary to set stricter color and distance thresholds to avoid excessive extension of the arm length which leads to an increase in the error matching rate in the edge region. The window arm length constraints are updated as:

$$\begin{cases} G(p) > \beta \\ D_s(p_l, p) < \lambda_1 \cdot L_1 \\ D_c(p_l, p) < \tau_1, D_c(p_l, p_l + (1, 0)) < \tau_1 \\ D_c(p_l, p) < \lambda_3 \cdot \tau_2, \text{ if } \lambda_2 \cdot L_2 < D_s(p_l, p) < \lambda_1 \cdot L_1 \end{cases}, \quad (2.9)$$

where the coefficients λ_1 , λ_2 , and λ_3 are the reduction factors of L_1 , L_2 , and τ_2 , respectively. With the constrains mentioned above, the length of the left, right, upper, and lower arms $\{h_p^-, h_p^+, v_p^+, v_p^-\}$ can be determined by linear extension from center pixel p , pixel by pixel. Therefore, a cross composed of horizontal line segment $H(p)$ and vertical line segment $V(p)$ is constructed as the local window of cost aggregation. To speed up aggregation, we further use an orthogonal integral image (OII) technique [14] for fast cost aggregation over any arbitrarily shaped windows in constant time. First, we build a horizontal integral image $S^H(x, y)$ to store the cumulative row sum as:

$$S^H(x, y) = S^H(x - 1, y) + C(x, y). \quad (2.10)$$

The $S^H(x, y)$ can be calculated from $S^H(x - 1, y)$ with only one addition. When $x = 0, S^H(-1, y) = 0$, we calculate the horizontal integral $E^H(x, y)$, using the horizontal integral image $S^H(x, y)$ as follows:

$$E^H(x, y) = S^H(x + h_p^+, y) - S^H(x - h_p^- - 1, y). \quad (2.11)$$

Taking the calculated horizontal matching cost $E^H(x, y)$ as the input, we create a vertical integral image $S^V(x, y)$. It stores the cumulative column sum as:

$$S^V(x, y) = S^V(x, y - 1) + E^H(x, y). \quad (2.12)$$

Also, only one addition is needed to calculate $S^V(x, y)$. When $y = 0, S^V(x, -1) = 0$, the fully aggregated matching cost $\tilde{C}(x, y)$ can be derived from the following equation:

$$\tilde{C}(x, y) = S^V(x, y + v_p^+) - S^V(x, y - v_p^- - 1). \quad (2.13)$$

Then, the error matching rate is further decreased by scan-line optimization in four directions.

Algorithm 2: Pseudocode for ACP method

Data: The initial image I^s , different scales of initial cost volume C^s , $s \in \{0, 1, \dots, S\}$.

Result: Final cost volume \tilde{C}^s .

```

1 for  $s = 0$  to  $S$  do
2   for  $y = 1$  to  $I^s.height$  do
3     for  $y = 1$  to  $I^s.width$  do
4       Compute  $\tau_2$  using Eq (2.8) and compute four arm lengths  $\{h_p^-, h_p^+, v_p^+, v_p^-\}$  for pixel  $p$ 
         in  $I^s$  using Eqs (2.7) and (2.9).
5       for  $m = 0$  to  $x$  do
6          $S^H(x, y) = S^H(m - 1, y) + C^s(m, y)$ ;
7       end
8        $E^H(x, y) = S^H(x + h_p^+, y) - S^H(x - h_p^- - 1, y)$ ;
9       for  $n = 0$  to  $y$  do
10         $S^V(x, y) = S^V(x, n - 1) + E^H(x, n)$ ;
11      end
12       $\tilde{C}^s(x, y) = S^V(x, y + v_p^+) - S^V(x, y - v_p^- - 1)$ ;
13    end
14  end
15 end

```

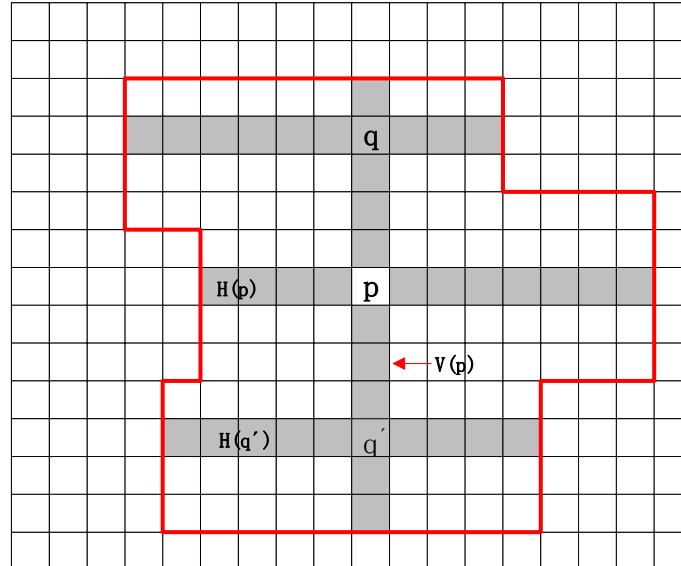


Figure 5. The cross-based support window region at each pixel. $V(p)$ and $H(p)$ are the vertical and horizontal segments of each pixel, respectively.

Figure 6 shows an example of the support window generated by the improved cost aggregation. It can be found that our support window includes fewer invalid pixels in the discontinuous depth region, which reduces the error matching rate.



Figure 6. Examples of support windows on the Monopoly image. (a) Adaptive window based on the cross proposed by Mei et al. [13]. (b) Improved cross adaptive window.

2.3. Disparity computation and disparity refinement

After cost aggregation, the cost volume is rectified. It is followed by disparity computation. The initial disparity of a pixel is computed by WTA strategy whose implementation is as follows: The corresponding disparity of the smallest cost value within the maximum disparity L_{\max} search range is taken as the initial disparity value. The process can be expressed as follows:

$$d_{ini}(p) = \arg \min_{d \in D} [S(p, d)], \quad (2.14)$$

where D is the disparity searching range, $d_{ini}(p)$ is the initial disparity of pixel p , and $S(p, d)$ is the matching cost after the cost aggregation. Generally, the minimum cost is selected to represent the optimal matching relationship. During disparity refinement, the outliers are picked up by left-right consistency detection. If the difference of disparity value between the pixel whose disparity is in the left image and the corresponding matching pixel whose disparity is in the right image satisfies the constraint:

$$|d_1 - d_2| > Th, \quad (2.15)$$

where Th is a default disparity threshold, generally set to 1, then $p(x, y)$ is treated as an anomaly point for subsequent processing or will be retained. To correct the anomaly point p , two non-anomaly points, p_l and p_r , are picked up on the left and right directions of point p , respectively. The minimum disparity value between p_l and p_r is selected as the corrected disparity value of p , which can be expressed by the following equation:

$$d(p) = \min(d(p_l), d(p_r)). \quad (2.16)$$

As there may be random noise and discontinuous error matching pixel in the disparity map, the median filter is used to smooth the disparity map, and consequently a high-quality disparity map is obtained.

3. Experiments

The experimental environment: C++ Compiler with OpenCV3.4 library, Windows 10 \times 86 operating system, Intel(R) Core(TM) i5-7200U CPU and 4 GB RAM, and Middlebury stereo datasets [26] were used to evaluate the proposed iFCTACP. In the following experiments, the best results among the compared algorithms are marked in bold. In this paper, the percentage of bad pixels in the estimated stereo pair difference is used as the evaluation criterion. The disparity error threshold is set to 1.0 pixel. The error matching rate is calculated for non-occluded regions by dividing the sum of all the mismatch points by the sum of the overall effective disparity points. The parameters used in the experiments are shown in Table 1.

Table 1. Parameters used in the experiments.

Parameters	α	τ_{AD}	τ_{Grad}	λ_{cen}	λ'	L_1	L_2
Values	0.1	7	2	25	700	34	7
Parameters	τ_1	τ_2	β	λ_1	λ_2	λ_3	λ_L
Values	20	6	0.6	0.15	0.25	0.5	3

3.1. Evaluation of the robustness to noise of the cost computation

We use six sets of images with ground truth provided by the Middlebury dataset [26] (Art, Dolls, Flowerpots, Moebius, Reindeer, and Cloth1) and set three illuminations (0, 1, 2) and three exposures (0, 1, 2). For simplicity, we only take the Art set as an example hereafter. Figure 7 shows the left images of Art under three different illuminations (the exposure is the same) and the left images of Art under three different exposures (the illumination is the same). In order to verify the effectiveness and noise robustness of iFCT, the iFCT and the other four transforms, CT, MCT, SCT, and MRCT, are respectively fused to the cost calculation method, which is composed of the truncated color absolute difference and the truncated gradient absolute difference for the comparative experiment. To verify the superiority of iFCT, all disparity maps are produced with the same cost aggregation method.

Figures 8 to 11 show the disparity map in four kinds of conditions: different illumination, different exposure, 5% salt-and-pepper noise, and Gaussian noise with $\sigma = 5$, respectively. In these figures, (a)–(c) show the left image of Art, the right image of Art, and the ground truth disparity map, respectively; (d)–(h) show the disparity map obtained by the cost computation method, which uses the truncated color absolute difference and the truncated absolute gradient difference to fuse five different census transform of iFCT, CT, MCT, SCT, and MRCT, respectively. In Figure 8, the left and right image are taken under illumination 2 and illumination 1, respectively, both under exposure 0. In Figure 9, the left image is taken under the condition of exposure 1, the right image is taken under the condition of exposure 2, and both images are taken under illumination 0. In Figure 10, 5% salt-and-pepper noise is added to the left and right images. In Figure 11, Gaussian noise with $\sigma = 5$ is added to the left and right images. Qualitatively, comparing Figure 8(d) with Figure 8(e)–(h), shows that the fused iFCT is more robust to image illumination changes. Also, comparing Figure 9(d) with Figure 9(e)–(h), the disparity map of Figure 9(d) show fewer artifacts than those of Figure 9(e)–(h). Comparing Figure 10(d) with Figure 10(e)–(h), it is found that Figure 10(e), which is added 5% salt-and-pepper noise, performs better than Figure 10(d), which is obtained by fusing iFCT. The disparity maps obtained by fusing the

above five different census transforms under the condition of Gaussian noise with the $\sigma = 5$ are shown in Figure 11, and showing the effect of the disparity map in Figure 11(f) outperforms the others.

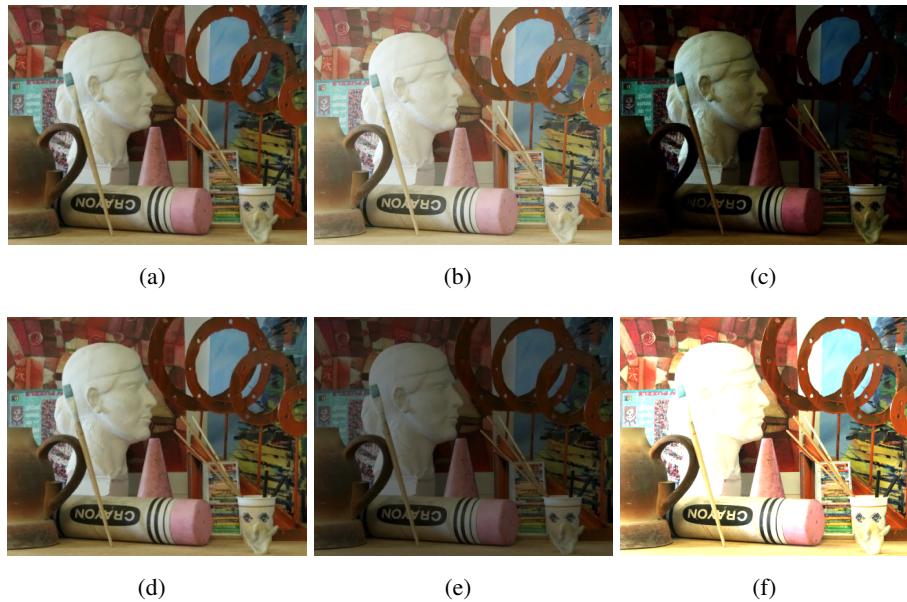


Figure 7. First row: left images of three different illuminations (under the same exposure 0). (a) Illumination 0. (b) Illumination 1. (c) Illumination 2. Second row: left images of three different exposures (under the same illumination 0). (d) Exposure 0. (e) Exposure 1. (f) Exposure 2.

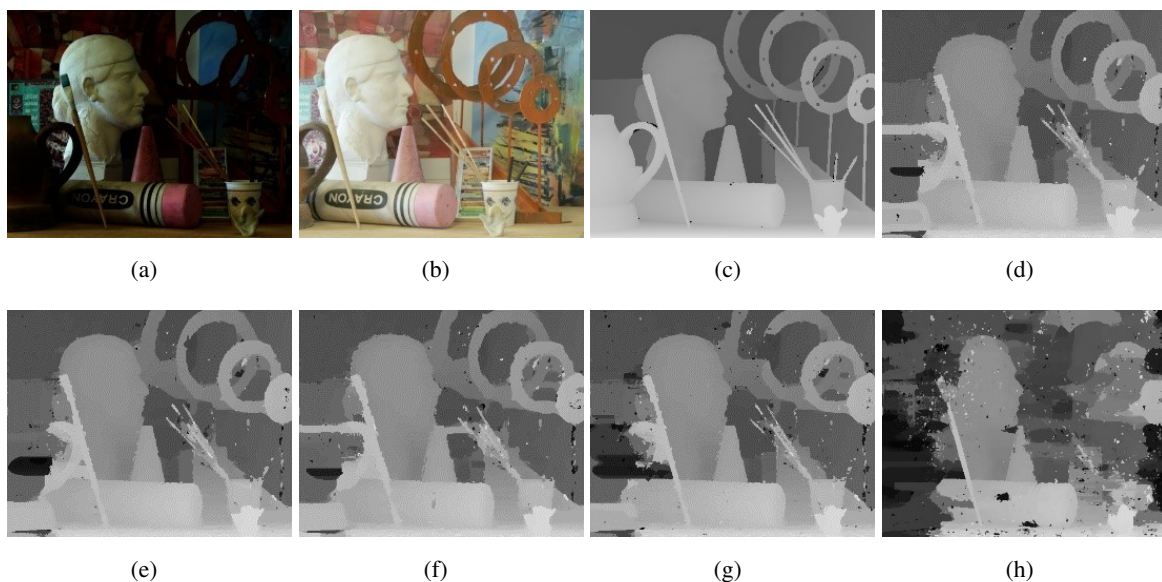


Figure 8. Disparity map of Art under different illumination conditions. (a) Left image under illumination 2. (b) Right image under illumination 1. (c) Ground truth disparity map. (d) iFCT. (e) CT. (f) MCT. (g) SCT. (h) MRCT.

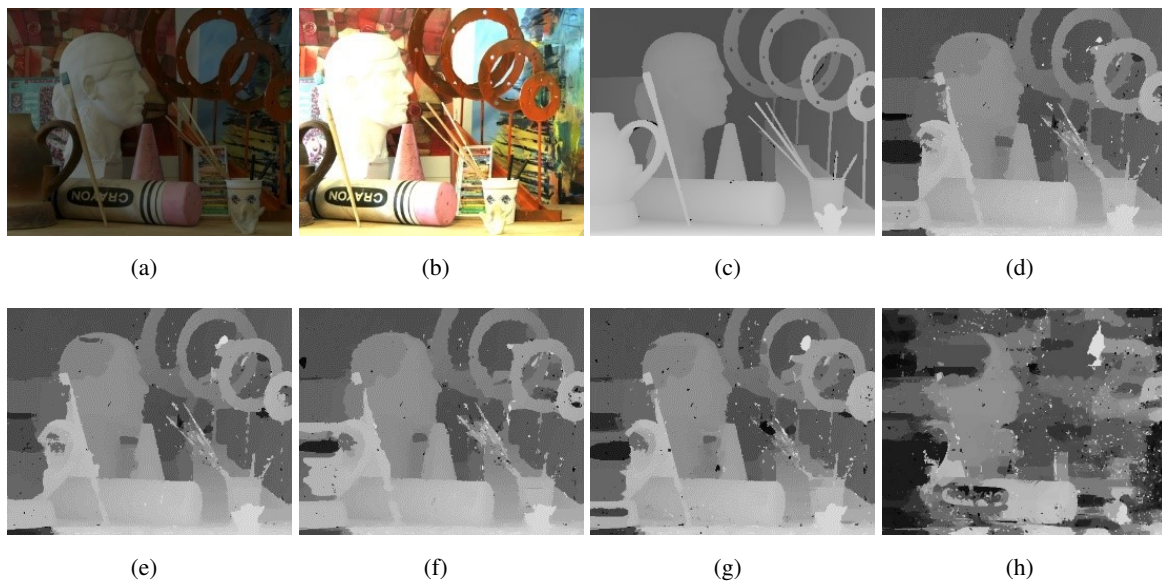


Figure 9. Disparity map of Art under different exposure conditions. (a) Left image under exposure 1. (b) Right image under exposure 2. (c) Ground truth disparity map. (d) iFCT. (e) CT. (f) MCT. (g) SCT. (h) MRCT.

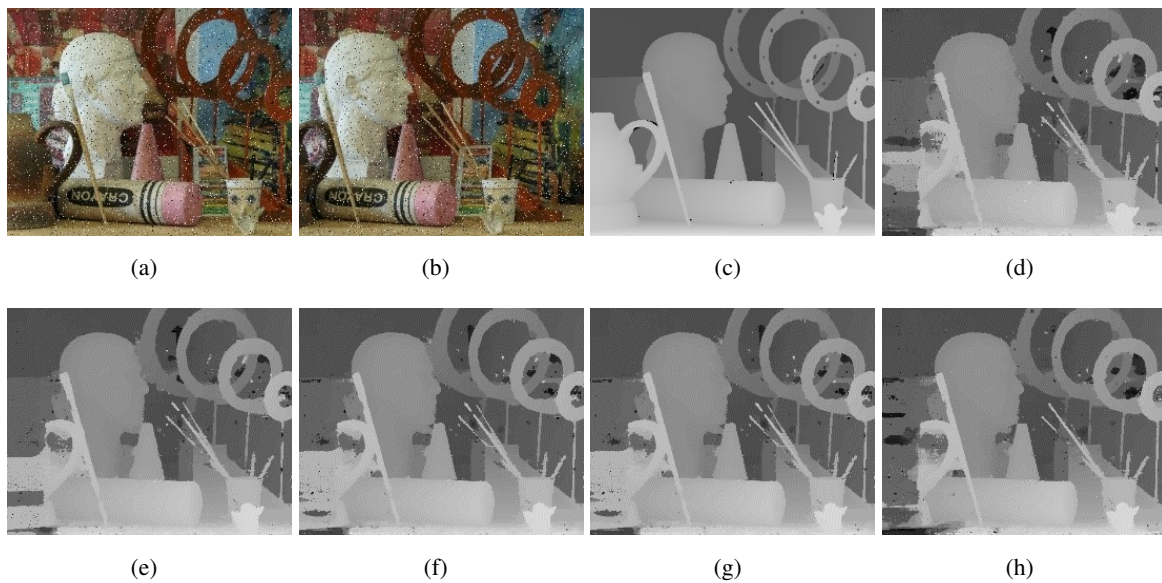


Figure 10. Disparity map of Art under 5% salt-and-pepper noise. (a) Left image under 5% salt-and-pepper noise. (b) Right image under 5% salt-and-pepper noise. (c) Ground truth disparity map. (d) iFCT. (e) CT. (f) MCT. (g) SCT. (h) MRCT.

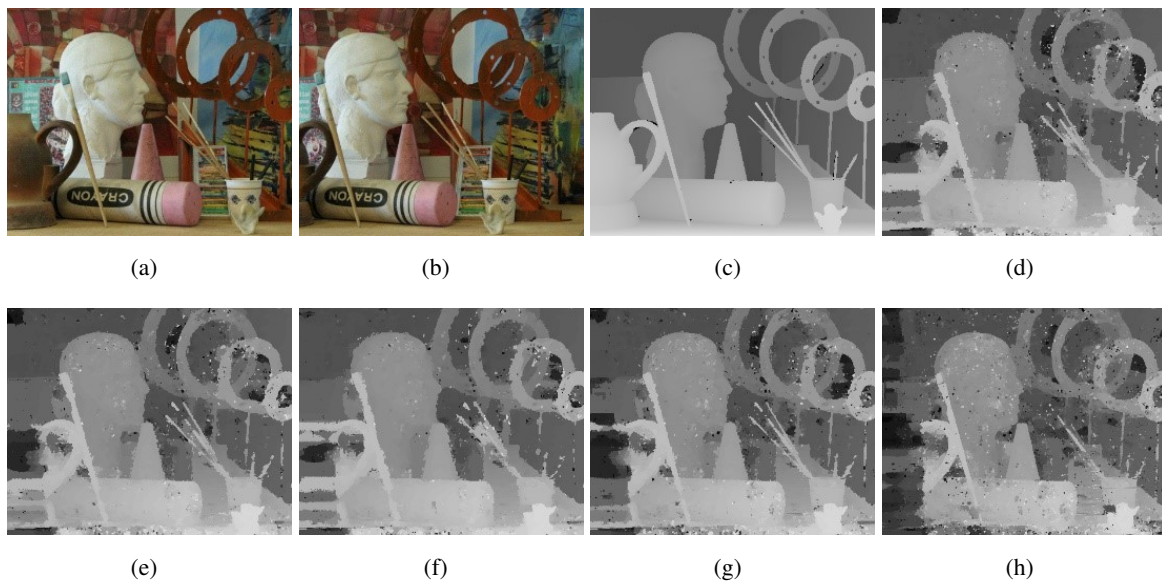


Figure 11. Disparity map of Art under the condition of Gaussian noise with $\sigma = 5$. (a) Left image with Gaussian noise ($\sigma = 5$). (b) Right image with Gaussian noise ($\sigma = 5$). (c) Ground truth disparity map. (d) iFCT. (e) CT. (f) MCT. (g) SCT. (h) MRCT.

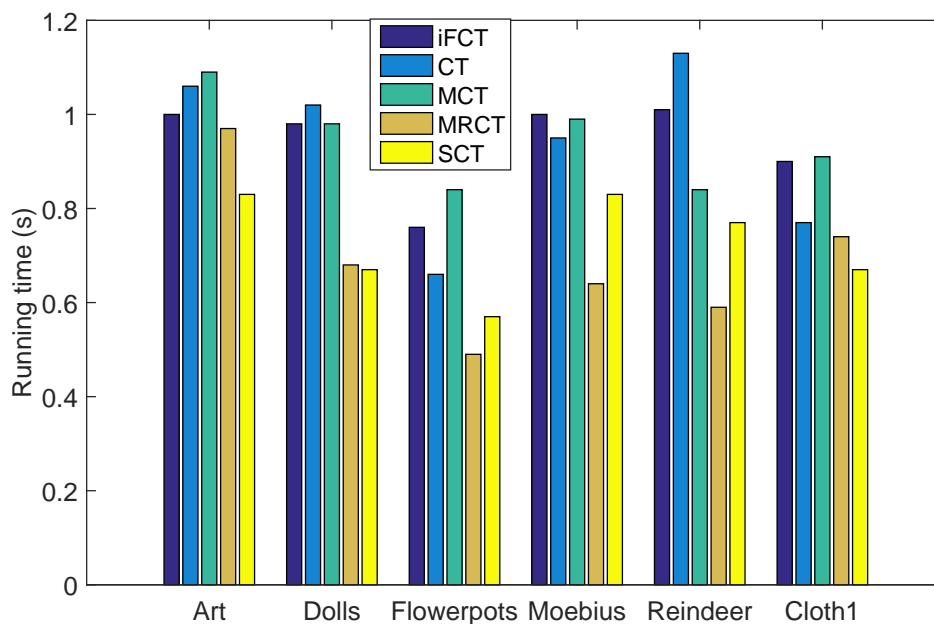


Figure 12. Running time of our algorithm on different images (in seconds).

Under the four conditions above, the truncated color absolute difference and truncated gradient absolute difference fuse those five different census transforms mentioned above. The error matching rates of the disparity maps are shown in Tables 2 to 5. For comparison, the disparity map without any noise is also measured in this paper, and the results are shown in Table 6. In Tables 2 and 3, among six pairs of stereo images, those fused with iFCT have the lowest error matching rate and the lowest

average error matching rate, which indicates that the iFCT is more robust to illumination change and exposure change than the other four census transforms. Table 4 shows, under the condition of 5% salt-and-pepper noise, that the average error matching rate of the fused CT is the lowest and that of the fused iFCT is at the middle level. In Table 5, the fused iFCT has the lowest error matching rate in 3 pairs of stereo images: Art, Reindeer, and Cloth1. The average error matching rate is only 0.1% than the optimal MCT, indicating that the fused iFCT is more robust to Gaussian noise. In Table 6, the fused iFCT achieves the best average error matching rate, which indicates that the iFCT is effective for noise-free matching. From the above analysis, we can see that iFCT exhibits excellent performance under illumination changes, exposure changes, and free noise. It also has good noise immunity to salt-and-pepper noise and Gaussian noise. In order to test the real-time performance of the algorithm, we test the execution time of it. The measured time is the average of 20 separate runs and the results are shown in Figure 12.

Table 2. Error matching rate of various cost computations under different illumination conditions.

Algorithms	Art	Dolls	Flowerpots	Moebius	Reindeer	Cloth1	Avg
iFCT	10.12	7.55	5.36	8.56	10.09	0.36	7.00
CT	10.53	9.21	5.60	8.92	10.59	0.37	7.54
MCT	12.34	9.84	5.99	9.28	10.86	0.42	8.12
SCT	14.13	11.60	6.94	10.68	15.80	0.54	9.95
MRCT	44.4	39.46	8.66	33.08	60.21	14.31	33.36

Table 3. Error matching rate of various cost computations under different exposure conditions.

Algorithms	Art	Dolls	Flowerpots	Moebius	Reindeer	Cloth1	Avg
iFCT	14.35	7.25	14.85	9.94	10.77	0.63	9.63
CT	14.99	7.72	15.20	10.99	10.74	0.68	10.05
MCT	16.69	8.88	16.78	12.26	12.80	0.79	11.37
SCT	18.52	13.29	22.45	13.45	13.50	1.26	13.75
MRCT	60.45	61.64	59.90	49.78	62.04	22.12	52.66

Table 4. Error matching rate of various cost computations under the condition of 5% salt-and-pepper noise.

Algorithms	Art	Dolls	Flowerpots	Moebius	Reindeer	Cloth1	Avg
iFCT	7.67	3.72	7.21	7.42	3.68	0.41	5.01
CT	6.74	3.14	6.51	7.21	3.38	0.41	4.57
MCT	6.77	4.98	11.81	10.47	6.63	0.34	6.83
SCT	6.77	3.06	8.17	7.43	3.28	0.36	4.85
MRCT	7.77	3.48	9.37	8.90	3.93	1.00	5.74

Table 5. Error matching rate of various cost computations under Gaussian noise with $\sigma = 5$.

Algorithms	Art	Dolls	Flowerpots	Moebius	Reindeer	Cloth1	Avg
iFCT	17.90	12.34	33.54	16.84	24.35	0.47	17.57
CT	19.83	13.12	33.61	17.30	25.40	0.53	18.30
MCT	18.55	11.90	32.81	16.53	24.68	0.48	17.49
SCT	22.30	14.86	35.82	19.10	29.78	0.78	20.44
MRCT	35.06	23.87	48.78	30.23	45.88	5.95	31.63

Table 6. Error matching rate of various cost computations under no-noise condition.

Algorithms	Art	Dolls	Flowerpots	Moebius	Reindeer	Cloth1	Avg
iFCT	5.81	2.88	5.36	5.60	3.34	0.24	3.87
CT	6.07	2.69	5.60	5.89	3.12	0.23	3.93
MCT	7.59	3.35	5.99	6.74	4.17	0.24	4.68
SCT	6.06	2.69	6.94	6.39	3.11	0.26	4.24
MRCT	6.80	2.96	8.66	8.90	3.55	0.63	5.25

3.2. Evaluation of cost aggregation

To verify the matching accuracy in weak texture regions, six groups of typical weak texture images, Lampshade1, Lampshade2, Midd1, Midd2, Monopoly, and Plastic, are selected from Middlebury for comparing the cost aggregation method based on ACP with four classical methods, AD-Census [13], GF [25], PPEP-GF [27], and Z2ZNCC [28]. All of them use the same cost computation method, iFCT, which was shown to be more robust than other combined cost computation methods in Section 3.1. The stereo matching results of six pairs of weak texture stereo images are shown in Figure 13, and the error matching points are marked in red. Comparing the disparity map of ACP in Figure 13(c) with those of the four classical methods (AD-Census, GF, PPEP-GF, Z2ZNCC) in Figure 13(d)–(g), it can be concluded that the ACP method is smoother in weak texture regions, the edges are better maintained, the obtained disparity maps are generally of high quality, and the error matching rate is much lower than that of the other classical methods.

The matching results and percentage of bad pixels are shown in Table 7. ACP cost aggregation method is applied to six pairs of weak texture stereo images, of which five pairs (Lampshade1, Lampshade2, Midd2, Monopoly, Plastic) show more remarkable results than the four classical cost aggregation methods. In addition, except for the stereo matching images, Plastic and Midd2, the mismatching rates of the other three pairs are less than 4%. Compared with AD-Census, GF, PPEP-GF, and Z2ZNCC, the aggregation error of ACP is reduced by 5.58%, 7.64%, 2.49%, and 6.68%, respectively. From the visual comparison of the original disparity map in Figure 13(b), it can be found that AD-Census performs worse than ACP and Z2ZNCC in the weak texture regions, because AD-Census adopts the same matching cost aggregation method in both the weak texture regions and the edge areas with rich textures and fails to utilize the regional characteristics of pixels in the image. GF and PPEP-GF have high mismatching rate in the weak texture region, because the linear coefficient of the guide filter is affected by the selected regularization parameters, the cost aggregation effect is not ideal, and the accuracy of the final disparity map is affected. On the whole,

the cost aggregation method of ACP obviously outperforms the four classical cost aggregation methods in weak texture regions and obtains disparity maps with a more correct area. The ACP achieves the lowest average error matching rate. At the same time, the running time of the above five cost aggregation methods is measured, as shown in Figure 14. To achieve a fair comparison, all of the algorithms are run on the same testing platform, and no parallelism technique is used. Seen from Figure 14, it takes both the AD-Census and ACP methods approximately 4 seconds. The GF method takes a little longer, while the PPEP-GF and Z2ZNCC methods take a little less time.

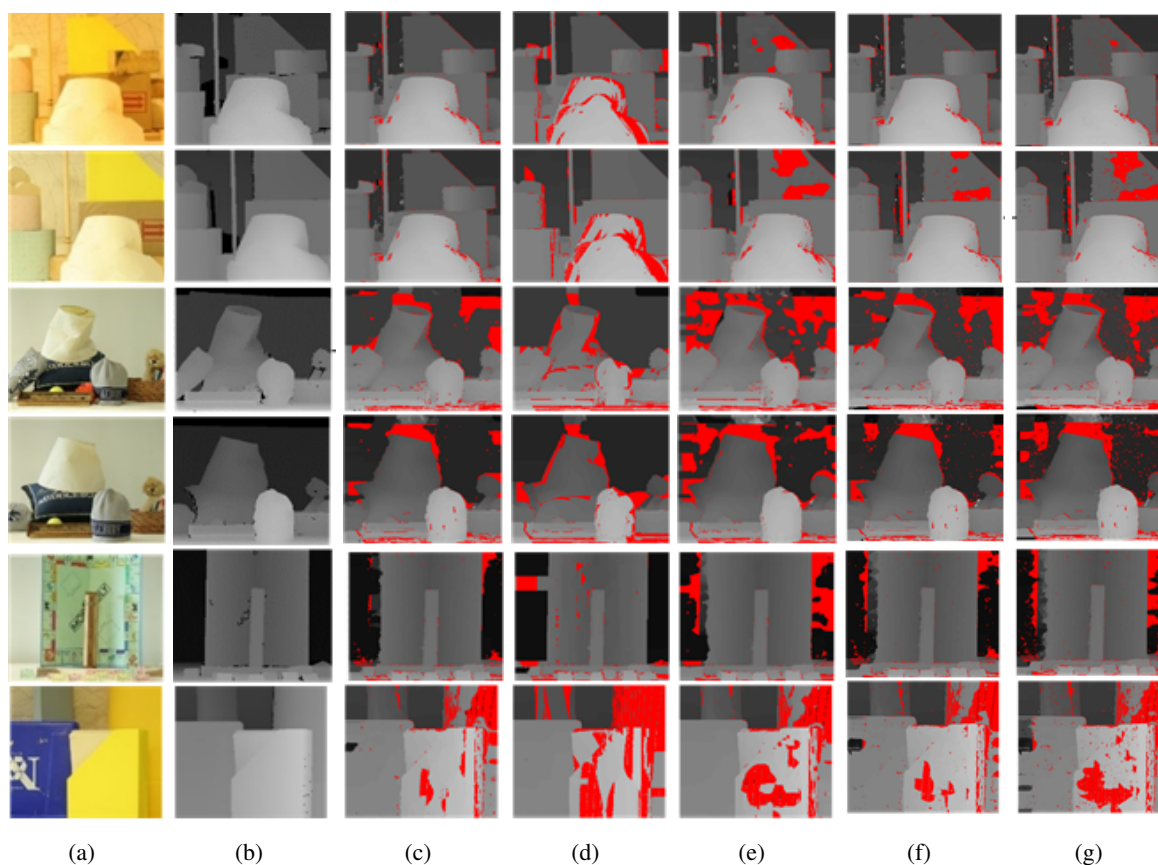
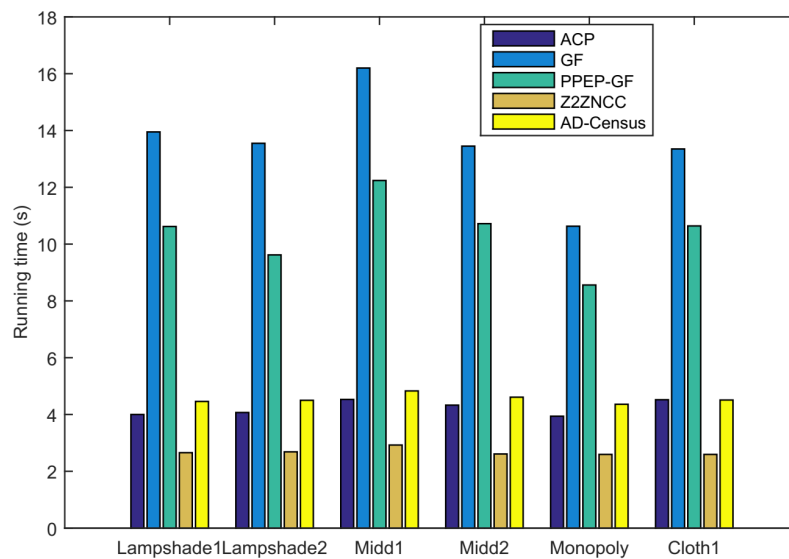


Figure 13. The comparisons of cost aggregation method based on ACP with four classical methods using Middlebury dataset. From top to bottom: Lampshade1, Lampshade2, Middle1, Middle2, Monopoly, Plastic. (a) Left image. (b) Ground truth disparity map. (c) Results of the ACP cost aggregation. (d) Results of ad-census cost aggregation. (e) Results of GF cost aggregation. (f) Results of PPEP-GF cost aggregation. (g) Results of Z2ZNCC cost aggregation.

Table 7. Error matching rate of low texture image.

Algorithms	Lampshade1	Lampshade2	Midd1	Midd2	Monopoly	Plastic	Avg
ACP	2.41	3.12	14.34	10.53	3.95	14.58	8.12
AD-census	10.34	10.69	11.82	12.35	5.44	32.07	13.79
GF	4.66	9.89	26.54	19.28	12.01	22.16	15.76
PPEP-GF	2.58	6.24	18.79	13.09	6.85	16.13	10.61
Z2ZNCC	3.12	9.56	24.12	17.75	11.44	22.80	14.8

**Figure 14.** Running time of our algorithm on different images (in seconds).

3.3. Comparative analysis of algorithms

To objectively evaluate the overall performance of the iFCTACP, the experiments were carried out on 25 pairs of stereo images from the Middlebury dataset. It was not only compared with the AD-census algorithm [19] but also with other recently proposed methods: the AEGF [29], OLT [30], and DDL algorithms [31]. The quantitative evaluation results are shown in Table 8. Compared with the other four algorithms, the IFCTACP algorithm has the lower average error rate and better effect on most stereo images. As shown in the paper, 17 out of 25 pairs of stereo images have the lowest error matching rate. Compared with AD-Census algorithm, AEGF algorithm, OLT algorithm and DDL algorithm, the average matching error of ACP algorithm in this paper is reduced by 4.97%, 2.25%, 2.09%, and 1.34%, respectively, showing more stable performance in processing different scenarios. This is due to the fact that DDL learns a discriminant dictionary for stereo matching and has non-local aggregation capability, which can cope well with low-texture regions, but is less effective for high-texture regions. AEGF and OLT do not perform as well as IFCTACP in low-texture areas because their cascaded filters only use fixed-size windows and have no spatial adaptability. The AD-Census algorithm is not ideal when determining the support domain, and the matching accuracy is not high in the weak texture region. In contrast, the parallax map obtained by iFCTACP not only contains less noise, but also has better performance in the weak texture region and high texture region.

Table 8. Error matching rate of different algorithms on Middlebury standard stereo image pairs.

Datasets	AD-census	AEGF	OLT	DDL	iFCTACP
Cones	8.61	2.01	3.42	4.27	2.96
Teddy	6.48	5.21	7.69	6.59	7.45
Tsukuba	8.04	1.32	2.06	2.63	3.92
Venus	2.92	0.26	0.54	0.69	1.34
Aloe	5.18	5.25	2.91	5.28	2.83
Art	11.58	7.58	6.91	9.47	5.81
Baby1	5.56	3.18	4.03	2.78	1.79
Baby2	8.27	4.15	4.86	2.84	1.89
Baby3	4.21	4.34	4.89	3.19	2.24
Books	11.05	8.05	8.07	8.16	6.87
Bowling1	18.77	13.33	14.52	4.38	3.12
Bowling2	13.27	7.13	6.11	4.43	2.69
Cloth1	0.20	0.49	0.11	0.25	0.24
Cloth2	4.15	2.88	1.30	2.14	1.17
Cloth3	1.68	1.31	0.93	1.80	0.93
Cloth4	1.32	1.54	0.74	1.21	0.77
Dolls	5.61	5.36	3.79	5.58	2.88
Flowerpots	17.26	5.36	11.35	6.80	5.36
Lampshade1	12.27	11.43	8.27	5.66	2.41
Lampshade2	15.23	16.98	12.20	4.08	3.12
Laundry	14.71	14.32	11.98	12.00	11.92
Moebius	9.97	8.71	8.25	9.60	5.60
Reindeer	16.70	3.57	4.11	5.40	3.34
Rocks1	2.54	3.25	1.27	4.14	1.40
Rocks2	2.04	2.44	1.02	3.26	1.08
Average error	8.30	5.58	5.42	4.67	3.33

3.4. Evaluation of the impact of parameter setting

If the accuracy of the initial disparity image is higher, the accuracy of the final disparity image will also be higher. In order to improve the accuracy of the initial disparity image, it is very important to select appropriate parameters for local stereo matching. Therefore, we will discuss the influence of the selection of different parameters on stereo matching. The average percentage of bad pixels under different parameters is used as the evaluation criteria.

The factors affecting cost computation are α , τ_{AD} , τ_{Grad} , λ_{cen} , and λ' . Among them, α plays an important role in adjusting the proportion of color intensity and gradient cost in the combined cost function; the color intensity threshold τ_{AD} and gradient threshold τ_{Grad} can effectively suppress some noise interference; and λ_{cen} and λ' control the proportion of iFCT cost and the combination cost of truncated color absolute difference and truncated gradient absolute difference in the fusion cost, respectively. In this paper, seven sets of images are selected as test images and the parameters are

adjusted separately, while other parameters are kept constant. The quantitative relationships between α , τ_{AD} , τ_{Grad} , λ_{cen} , λ' and the error matching rate are given in Figure 15(a)–(e) in detail. From Figure 15(a) to Figure 15(d), it can be concluded that the error matching rate of the disparity image increases when the parameter α increases, especially for the weak texture images, Plastic and Monopoly, which increases sharply. When $\tau_{Grad} \geq 2$, most error matching rates of the images do not vary obviously, except for Plastic, whose error matching rate is still increasing. The error matching rates change little when λ_{cen} ranges from 20 to 40; the weak texture images, Plastic and Monopoly, are highly sensitive to the parameter λ' , whose error matching rate decreases sharply when λ' increases. However, the other rich textured images are insensitive to λ' . From the experiments, we conclude that the overall performance is relatively better when $\alpha = 0.1$, $\tau_{AD} = 7$, $\tau_{Grad} = 2$, $\lambda_{cen} = 25$, and $\lambda' = 700$.

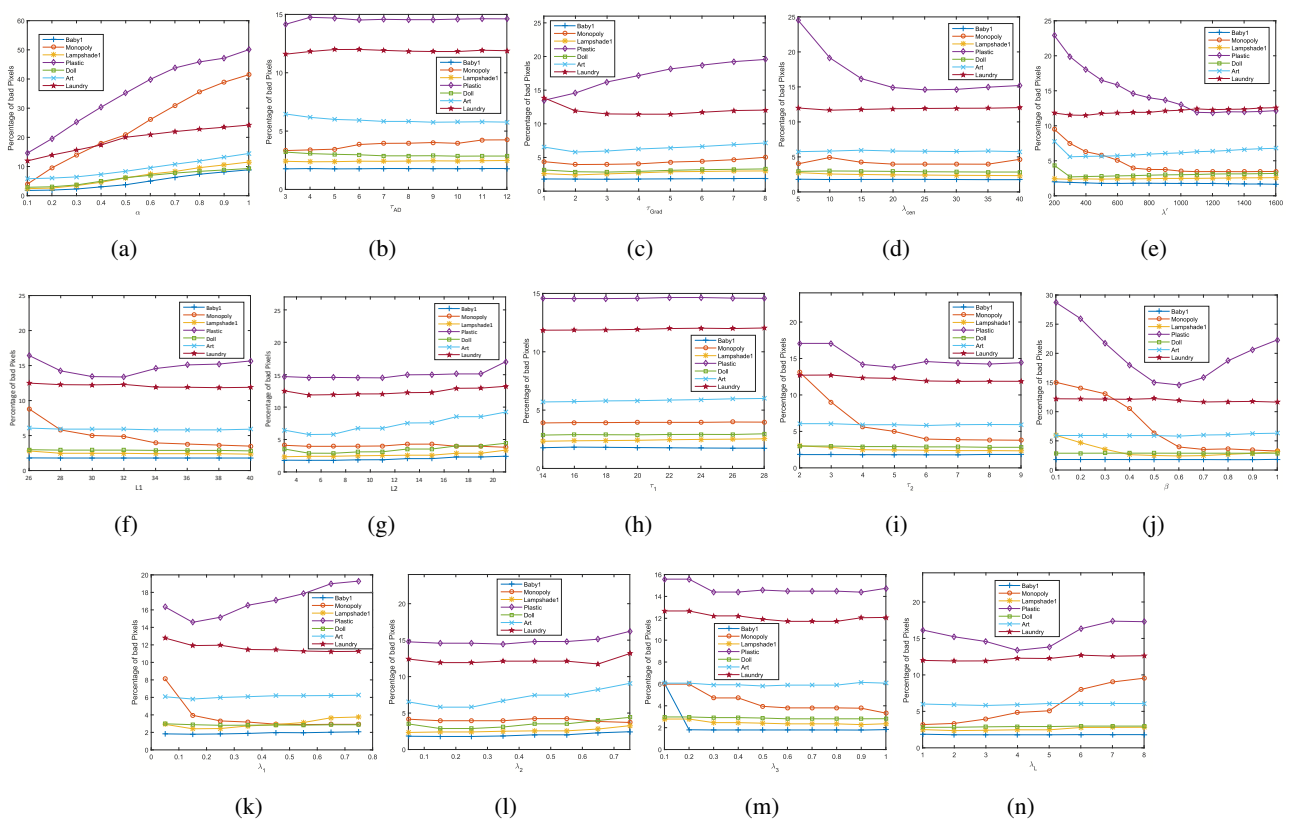


Figure 15. Experimental results of different parameter settings with Middlebury images. (a) α : Adjusting the proportion of color intensity and gradient cost in the combined cost function. (b) τ_{AD} : Color intensity threshold. (c) τ_{Grad} : Gradient cost threshold. (d) λ_{cen} : Improved census regularization parameters. (e) λ' : Regularization parameters of combined cost function. (f) Arm length threshold L_1 . (g) Arm length threshold L_2 . (h) Color similarity threshold τ_1 . (i) Color similarity threshold τ_2 . (j) β : Gradient threshold. (k) λ_1 : Scale factor of arm length threshold L_1 . (l) λ_1 : Scale factor of arm length threshold L_1 . (m) λ_1 : Scale factor of color similarity threshold τ_2 . (n) λ_L : Regularization parameters defined by color threshold.

During the construction of the cross-based supporting window, the arm length threshold L_1 and color similarity threshold τ_1 are used to retain more useful pixels in weak texture regions, which makes the result more accurate and robust. As shown in Figure 15(f),(h), the error matching rate tends to be stable when $L_1 \geq 34$, while τ_1 has little effect on it for most images. To avoid introducing more noise in the discontinuous disparity region when setting L_1 and τ_1 , a stricter color similarity threshold τ_2 is set when the arm length exceeds the threshold L_2 . In Figure 15(g), the parameter L_2 has little effect on the error matching rate of most images. In Figure 15(i), the error matching rate of all images does not change much when the parameter $\tau_2 \geq 6$. The gradient threshold β is set to determine whether the pixel point belongs to the weak texture region or the edge region. For the edge region, the scale factors λ_1 , λ_2 , and λ_3 need to be set for reducing the corresponding distance thresholds L_1 , L_2 , and color similarity threshold τ_2 . In Figure 15(j), β reduces the error matching rate for most of the images, especially for the weak texture images Monopoly. However, for Plastic, the error matching rate increases sharply when $\beta > 0.6$. Similarly, it can be seen from Figure 15(k) that when $\lambda_1 \geq 0.15$, the error matching rates tend to be stable for most images except for Plastic. Figure 15(l) that shows λ_2 reduces the error matching rate especially for Art images and the error matching rate goes up, as λ_2 increases. Figure 15(m) shows that the error matching rates of most images tend to be stable when $\lambda_3 \geq 0.5$. The λ_L is a variable color threshold. More constraints will be imposed on it when the arm length becomes longer. As shown in Figure 15(n), the error matching rates of the Plastic and Monopoly images fluctuate while those of other images are nearly kept unchanged. In this paper, to obtain a better matching rate, we set $L_1 = 34$, $L_2 = 7$, $\tau_1 = 20$, $\tau_2 = 6$, $\lambda_1 = 0.15$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.5$.

4. Conclusions

In order to improve the anti-noise robustness and obtain better disparity map in large weak texture regions, we propose a stereo matching algorithm, iFCTACP, which encompasses two innovative points. First, the iFCT, a model based on regional information coding, is used as a new and stable method for cost measurement, with the benefits of its strong noise resistance to illumination and exposure. Secondly, the adaptive cross pyramid model, ACP, can build connections among different regions with similar disparity to fully use image region features. The qualitative and quantitative experiments on the Middlebury dataset validate the effectiveness of iFCTACP, outperforming most known local stereo matching algorithms. Since the depth information can better distinguish between near and distant objects, the solution in this paper can be combined with other applications, such as 360 vision, to further improve the multi-target tracking performance.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgments

This research was funded by the 2023 Fujian Province Young and Middle-aged Teacher Education Research Project (Science and Technology), grant number JZ230076.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Liu, J. Gao, S. Ji, C. Zeng, S. Zhang, J. Gong, Deep learning based multi-view stereo matching and 3D scene reconstruction from oblique aerial images, *ISPRS J. Photogramm. Remote Sens.*, **204** (2023), 42–60. <https://doi.org/10.1016/j.isprsjprs.2023.08.015>
2. Y. Zhang, Y. Su, J. Yang, J. Ponce, H. Kong, When dijkstra meets vanishing point: A stereo vision approach for road detection, *IEEE Trans. Image Process.*, **27** (2018), 2176–2188. <https://doi.org/10.1109/tip.2018.2792910>
3. Y. Shi, Y. Guo, Z. Mi, X. Li, Stereo centerNet based 3D object detection for autonomous driving, *Neurocomputing*, **417** (2022), 219–229. <https://doi.org/10.1016/j.neucom.2021.11.048>
4. R. A. Hamzah, H. Ibrahim, Literature survey on stereo vision disparity map algorithm, *J. Sensors*, **2016** (2016), 8742920. <https://doi.org/10.1155/2016/8742920>
5. S. Ahn, M. Chertkov, A. E. Gelfand, S. Park, J. Shin, Maximum weight matching using odd-sized cycles: Max-product belief propagation and half-integrality, *IEEE Trans. Inf. Theory*, **64** (2017), 1471–1480. <https://doi.org/10.1109/tit.2017.2788038>
6. C. Shi, G. Wang, X. Yin, X. Pei, B. He, X. Lin, High-accuracy stereo matching based on adaptive ground control points, *IEEE Trans. Image Process.*, **24** (2015), 1412–1423. <https://doi.org/10.1109/tip.2015.2393054>
7. J. Cai, Integration of optical flow and dynamic programming for stereo matching, *IET Image Process.*, **6** (2012), 205–212. <https://doi.org/10.1049/iet-ipr.2010.0070>
8. M. Yang, F. Wang, Y. Wang, N. Zheng, A denoising method for randomly clustered noise in ICCD sensing images based on hypergraph cut and down sampling, *Sensors*, **17** (2017), 2778. <https://doi.org/10.3390/s17122778>
9. D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Comput. Vis.*, **47** (2002), 7–42. <https://doi.org/10.1023/A:1014573219977>
10. K. Y. Kok, P. Rajendran, A review on stereo vision algorithm: Challenges and solutions, *ECTI Trans. Comput. Inf. Technol.*, **13** (2019), 112–128. <https://doi.org/10.37936/ecti-cit.2019132.194324>
11. H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, *IEEE Trans. Pattern Anal. Mach. Intell.*, **30** (2007), 328–341. <https://doi.org/10.1109/tpami.2007.1166>
12. F. Stein, Efficient computation of optical flow using the census transform, in *Joint Pattern Recognition Symposium*, Springer, Berlin, Heidelberg, **3175** (2004), 79–86. https://doi.org/10.1007/978-3-540-28649-3_10
13. X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, X. Zhang, On building an accurate stereo matching system on graphics hardware, in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, Barcelona, Spain, (2011), 467–474. <https://doi.org/10.1109/iccvw.2011.6130280>

14. K. Zhang, J. Lu, G. Lafruit, Cross-based local stereo matching using orthogonal integral images, *IEEE Trans. Circuits Syst. Video Technol.*, **19** (2009), 1073–1079. <https://doi.org/10.1109/tcsvt.2009.2020478>
15. A. Hosni, M. Bleyer, M. Gelautz, Secrets of adaptive support weight techniques for local stereo matching, *Comput. Vis. Image Underst.*, **117** (2013), 620–632. <https://doi.org/10.1016/j.cviu.2013.01.007>
16. K. Zhang, Y. Fang, D. Min, L. Sun, S. Yang, S. Yan, et al., Cross-scale cost aggregation for stereo matching, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Columbus, USA, (2014), 1590–1597. <https://doi.org/10.1109/cvpr.2014.206>
17. Y. Pang, C. Su, T. Long, Adaptive multi-scale cost volume construction and aggregation for stereo matching (in Chinese), *J. Northeast. Univ. (Nat. Sci.)*, **44** (2023), 457–468. <https://doi.org/10.12068/j.issn.1005-3026.2023.04.001>
18. Y. Bi, C. Li, X. Tong, G. Wang, H. Sun, An application of stereo matching algorithm based on transfer learning on robots in multiple scenes, *Sci. Rep.*, **13** (2023), 12739. <https://doi.org/10.1038/s41598-023-39964-z>
19. H. Wei, L. Meng, An accurate stereo matching method based on color segments and edges, *Pattern Recognit.*, **133** (2023), 108996. <https://doi.org/10.1016/j.patcog.2022.108996>
20. M. S. Hamid, N. A. Manap, R. A. Hamzah, A. F. Kadmin, Stereo matching algorithm based on deep learning: A survey, *J. King Saud Univ.-Comput. Inf. Sci.*, **34** (2022), 1663–1673. <https://doi.org/10.1016/j.jksuci.2020.08.011>
21. B. Lu, L. Sun, L. Yu, X. Dong, An improved graph cut algorithm in stereo matching, *Displays*, **69** (2021), 102052. <https://doi.org/10.1016/j.displa.2021.102052>
22. L. Ma, J. Li, J. Ma, H. Zhang, A modified census transform based on the neighborhood information for stereo matching algorithm, in *2013 Seventh International Conference on Image and Graphics*, IEEE, Qingdao, China, (2013), 533–538. <https://doi.org/10.1109/icig.2013.113>
23. X. Lai, X. Xu, L. Lv, Z. Huang, J. Zhang, P. Huang, A novel non-parametric transform stereo matching method based on mutual relationship, *Computing*, **101** (2019), 621–635. <https://doi.org/10.1007/s00607-018-00691-3>
24. J. Lee, D. Jun, C. Eem, H. Hong, Improved census transform for noise robust stereo matching, *Opt. Eng.*, **55** (2016), 063107. <https://doi.org/10.1117/1.oe.55.6.063107>
25. A. Hosni, C. Rhemann, M. Bleyer, C. Rother, M. Gelautz, Fast cost-volume filtering for visual correspondence and beyond, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2012), 504–511. <https://doi.org/10.1109/cvpr.2011.5995372>
26. D. Scharstein, C. Pal, Learning conditional random fields for stereo, in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Minneapolis, USA, (2007), 1–8. <https://doi.org/10.1109/cvpr.2007.383191>
27. Y. Fu, K. Lai, W. Chen, Y. Xiang, A pixel pair-based encoding pattern for stereo matching via an adaptively weighted cost, *IET Image Process.*, **15** (2021), 908–917. <https://doi.org/10.1049/ipr2.12071>

28. Q. Chang, A. Zha, W. Wang, X. Liu, M. Onishi, L. Lei, Efficient stereo matching on embedded GPUs with zero-means cross correlation, *J. Syst. Archit.*, **123** (2022), 102366. <https://doi.org/10.1016/j.sysarc.2021.102366>
29. S. Zhu, Z. Wang, X. Zhang, Y. Li, Edge-preserving guided filtering based cost aggregation for stereo matching, *J. Vis. Commun. Image Represent.*, **39** (2016), 107–119. <https://doi.org/10.1016/j.jvcir.2016.05.012>
30. W. Wu, H. Zhu, Q. Zhang, Oriented-linear-tree based cost aggregation for stereo matching, *Multimed. Tools Appl.*, **78** (2019), 15779–15800. <https://doi.org/10.1007/s11042-018-6993-2>
31. J. Yin, H. Zhu, D. Yuan, T. Xue, Sparse representation over discriminative dictionary for stereo matching, *Pattern Recognit.*, **71** (2017), 278–289. <https://doi.org/10.1016/j.patcog.2017.06.015>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)