electronics

MDPI

Article

Stereo Matching Method with Cost Volume Collaborative Filtering

Wenhuan Wu *, Xi Xu, Wenshu Wang and Haokun Zhang

School of Electrical and Information Engineering, Hubei University of Automotive Technology, Shiyan 442002, China; 202211188@huat.edu.cn (X.X.); 202111078@huat.edu.cn (W.W.); 202111109@huat.edu.cn (H.Z.)

* Correspondence: wuwenhuan@huat.edu.cn

Abstract: Aiming at the problem of matching ambiguity and low disparity accuracy at the object boundary in stereo matching, a novel stereo matching algorithm with cost volume collaborative filtering is proposed. Firstly, for each pixel, two support windows are built, namely a local cross-support window as well as a global support window for the whole image. Secondly, a new adaptive weighted guide filter with a cross-support window as a kernel window is derived, and it is used to locally filter the cost volume. In addition, a minimum spanning tree is constructed in the whole image window, and then the minimum spanning tree filter is used to globally filter the cost volume. The collaborative filtering of cost volume is realized by fusing the filtering results of the local filter and global filter, so that each pixel can not only receive the support of the neighboring pixels in the local adaptive window, but can also receive the effective support of other pixels in the whole image, thus effectively eliminating the matching ambiguity in different texture regions while maintaining the disparity edges. The experimental results show that the average matching error rate of our method on the Middlebury stereo images is 3.17%. Compared with the other state-of-the-art methods, our method has higher robustness and matching accuracy, the generated disparity maps are smoother, and the disparity edges are better preserved.

Keywords: stereo matching; cost volume filtering; adaptive weighted guide filtering; tree filtering

check for updates

Citation: Wu, W.; Xu, X.; Wang, W.; Zhang, H. Stereo Matching Method with Cost Volume Collaborative Filtering. *Electronics* **2024**, *13*, 3095. https://doi.org/10.3390/electronics13153095

Academic Editor: Beiwen Li

Received: 25 June 2024
Revised: 29 July 2024
Accepted: 31 July 2024
Published: 5 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The goal of stereo matching is to determine the correspondences of all pixels between a pair of stereo images, so as to perceive the depth information and three-dimensional structure information of the scene according to the disparity of the corresponding pixels. Due to low cost, easy implementation, and the ability to recover the dense depth information of the scene, it has numerous applications such as autonomous driving [1], 3D object detection [2], 3D reconstruction [3], etc. In addition, stereo matching is also a key technology in photogrammetry [4]. By utilizing stereo matching, more common points are identified, and dense 3D point clouds are also obtained. It can provide good support for the subsequent triangulation and 3D model generation. It is assumed that there is a pair of stereo images that are rectified. Among them, one image is the reference image and the other is the target image. The task of stereo matching is to find the corresponding pixel in the target image for each pixel of the reference image. Because the input image pair is rectified and their epipolar lines are horizontal, the search of pixel correspondences between stereo images is conducted along horizontal lines. The *disparity* is the difference between the horizontal coordinates of a pair of corresponding pixels. For the pair of stereo images in Figure 1, the left image is the reference image and the right image is the target image. The coordinates of pixel p in the left image are (x, y), and its true corresponding pixel in the right image is $p\prime$, whose coordinates are $(x\prime, y)$, so that the disparity d of pixel p is the difference between the horizontal coordinate of pixels p and p', i.e., $d = x - x'$. The

resultant output of stereo matching is a dense disparity map by finding all the correspondences between stereo images [5]. Because of the matching ambiguity problem, multiple candidate pixels in the target image are often very similar to the pixel to be matched, and there is occlusion, noise, or distortion in the images, so stereo matching is very challenging.

Stereo matching methods can be mainly divided into global methods, local methods, and deep-learning-based methods. The global method calculates the disparities of all pixels simultaneously. This can be achieved by minimizing an energy function defined on a Markov random field with a global optimization approach such as graph cuts [6], belief propagation [7], and dynamic programming [8]. Global methods generally have high matching accuracy. However, their computational complexity is very high due to the iterative nature of the underlying optimization process, so they are difficult to apply in situations with strict real-time requirements. The local methods calculate the disparity of each pixel one by one using the local "Winner Takes All (WTA)" optimization algorithm [5]. For each pixel, the WTA algorithm directly chooses the disparity associated with the minimum aggregation cost. In recent years, with the development of deep neural networks, stereo matching methods based on deep learning are proposed [9]. These kinds of methods are data-dependent and the training process of the network model is very time-consuming. It is noted that deep-learning-based methods cannot achieve good and stable performance when changing a scenario because of the generalization problem. Generally, compared with global methods and deep-learning-based methods, local methods are less time-consuming and can better meet the requirements of practical applications. Therefore, we mainly focus on how to generate more accurate disparity maps using a local stereo matching strategy in our work.

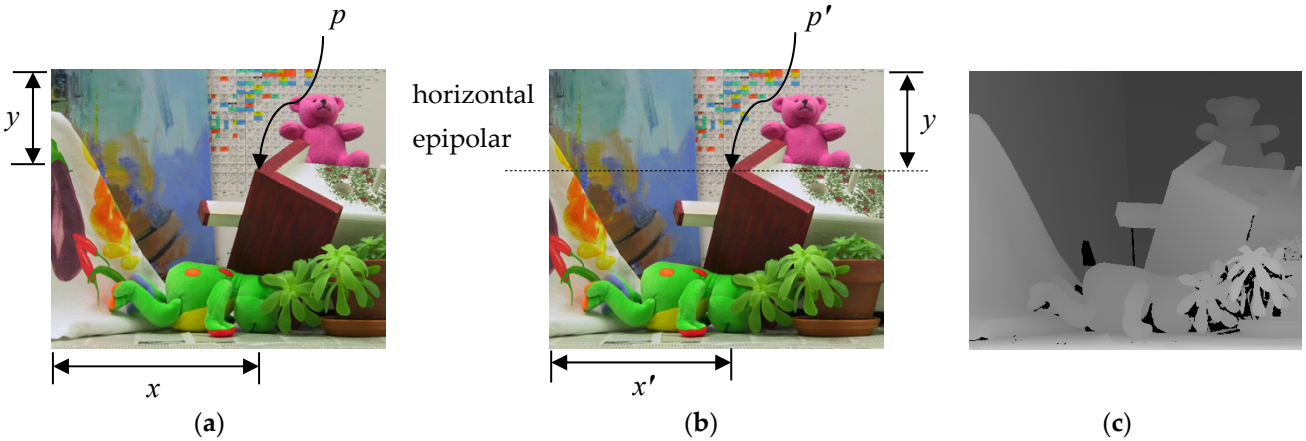


Figure 1. Stereo matching of Middlebury [10] *Teddy* image pair. (**a**) Reference (**left**) image; (**b**) target (**right**) image; (**c**) disparity map.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 systematically describes a novel stereo matching method based on cost volume collaborative filtering. Section 4 presents the experimental results and analyzes them. Section 5 concludes this paper.

2. Related Works

Stereo matching firstly needs to calculate the matching cost of each pixel at each disparity using some measurement of pixel similarity. Then, the raw matching costs of all pixels within a support window are aggregated to the center pixel at each disparity, aiming to eliminate matching ambiguity by increasing the cost distance between the real disparity and other candidate disparities. Here, there is an implicit assumption that the disparities of all pixels within a support window are similar. However, this assumption is not valid near depth discontinuities where the pixels in a support window come from different depths. This leads to the well-known edge-fattening effect. To resolve this problem, segmentation-

based methods [11,12] divide the image into different regions and take segmented regions as the support windows. However, these methods severely rely on segmentation accuracy and perform poorly for highly textured images. Zhang et al. [13] proposed to adaptively build cross-based support windows for each pixel, and adopted the orthogonal integral image technique to perform cost aggregation. However, this method does not preserve the disparity edges of objects well.

The adaptive support weight (ASW) methods assign an appropriate support weight to each pixel in a support window according to the proximity and color similarity to the center pixel. This is equivalent to segmenting the image in a soft way. Yoon and Kweon [14] computed the support weights of pixels using the bilateral filter. Hence, the cost aggregation essentially used the bilateral filter to smooth the cost volume. The bilateral filter can effectively preserve disparity edges, but its computational complexity is related to the size of the support window. Accordingly, the bilateral filter-based methods execute very slowly when the size of support windows is relatively large. To reduce computational complexity, several acceleration techniques [15,16] for stereo matching with bilateral filter have been proposed. However, these accelerated approximation techniques are often at the expense of accuracy.

Inspired by the work of [14], various edge-aware filtering methods have been introduced into stereo matching for better estimating support weights. Since the computational complexity of the guided filter (GF) [17] is independent of the kernel window size and has better performance than BF, Rhemann et al. [18] utilized GF to smooth the cost volume. Several improved GF-based methods (e.g., adaptive guided filtering [19] and cross-based local multi-point filtering [20]) combine GF with adaptive support windows to avoid the kernel windows covering different objects. To better preserve the object edges, Hamzah et al. [21] used a cascade model of BF and the iterative guided image filter (IGF) [22] to smooth the cost volume. Inspired by human beings, which process stereoscopic correspondence from coarse to fine, Zhang et al. [23] adopted the coarse-to-fine strategy and performed cross-scale cost aggregation by enforcing cost volume consistency across multiple scales. To reduce the computational complexity of cost volume filtering, Yuan et al. [24] presented the fast gradient domain guided image filter (F-GDGIF) with sub-sampling to filter cost volume. To improve the accuracy at disparity discontinuities, Fan et al. [25] presented a novel edge-preserving cost aggregation method by enforcing anisotropy.

To avoid designing an optimal support window for each pixel, stereo matching with recursive edge-aware filters [26,27] have been presented. However, when the recursive edge-aware filters are used for cost aggregation, each pixel can only receive support from pixels in the row or column direction and cannot directly receive support from adjacent pixels in other directions. Yang [28,29] proposed a non-local cost aggregation method by constructing a minimum spanning tree (MST), which has extremely low computational complexity. Mei et al. [30] proposed a segment-tree-based cost aggregation for stereo matching. Cheng et al. [31] proposed a cross-tree structure consisting of a horizontal tree and a vertical tree to perform cost aggregation. Compared to local ASW methods, which often take local fixed-size or adaptive regions as support windows, non-local methods directly take the whole image as the support window of each pixel. Non-local methods can better cope with low- or weak-texture regions, but they generally demonstrate poor performance in highly textured regions.

3. The Proposed Method

Stereo matching methods typically consist of four steps: (1) matching cost computation; (2) cost volume filtering; (3) disparity computation; and (4) disparity refinement. An overview of the proposed algorithm is shown in Figure 2. Below, the proposed algorithm will be described in detail following this pipeline.

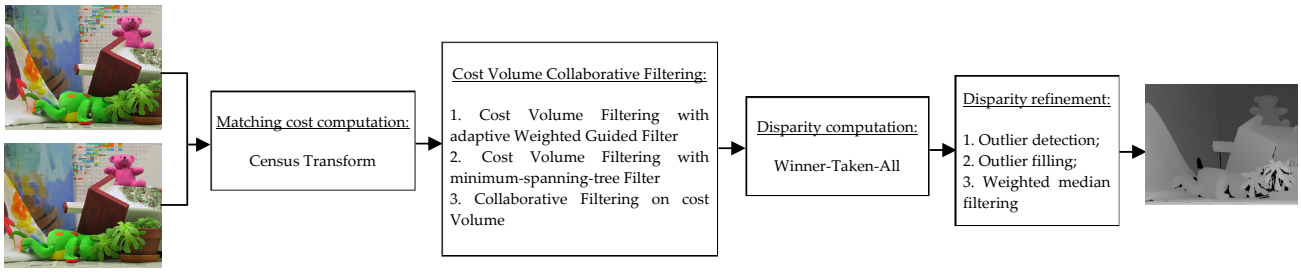


Figure 2. Schematic flow diagram of the proposed algorithm.

3.1. Matching Cost Computation

The matching cost represents the dissimilarity between the corresponding pixels in the left and right images. The census transform changes each pixel's intensity into a bit string. Because the bit string represents the relative ordering of the neighboring pixels, it can maintain invariance to radiation changes. The census transform shows better performance than the other matching cost calculation methods [32]. Hence, we calculate the matching cost with the census transform.

Supposing that $\mathcal{N}_p$ is a window centered on pixel p, the census transform of p is formulated as follows:

$$F(p) = \otimes_{q \in \mathcal{N}_p} \delta(I(p), I(q)) \tag{1}$$

where $F(p)$ is a bit string generated by the concatenation operation $\otimes$, and $\delta(i, j)$ is a binary function expressed as follows:

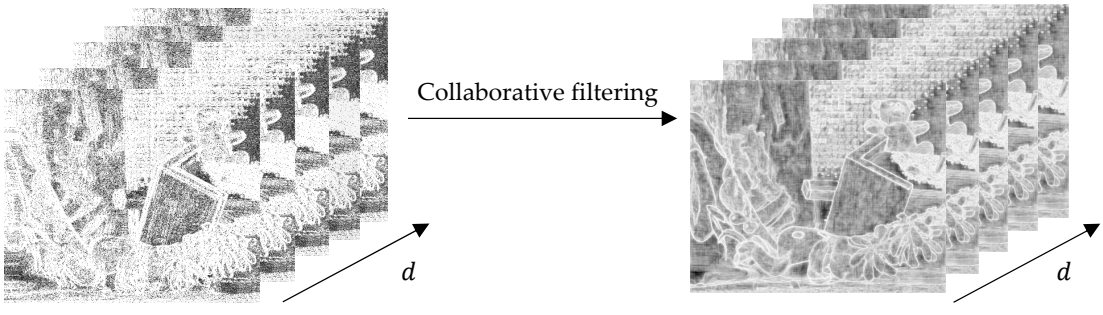
$$\delta(i, j) = \begin{cases} 1 & if\ i > j \\ 0 & else \end{cases} \tag{2}$$

It is assumed that the left image is taken as the reference image and $\mathcal{D}$ is the disparity level set. For pixel $p = (x, y)$ in the left image and an allowed disparity level of $d \in \mathcal{D}$, the corresponding pixel in the right image can be denoted as $p_d = (x - d, y)$. The matching cost $C(p, d)$ of pixel p with disparity d is defined as the Hamming distance of $F(p)$ and $F(p_d)$:

$$C(p, d) = h(F(p), F(p_d)) \tag{3}$$

where $h(u, v)$ denotes the Hamming function, which counts the number of different bits between the binary strings u and v.

As illustrated in Figure 3, after the matching costs $C(p, d)$ of each pixel with each allowed disparity level are computed, the initial cost volume C, which is a 3D array, can be constructed. As described in Section 3.2, each cost image of the initial cost volume C will be collaboratively filtered with the adaptive weighted guided filter and the minimum spanning tree filter. Then, the filtered cost volume C^A can be generated.



(**a**) (**b**)

Figure 3. The initial cost volume C and the filtered cost volume C^A. (**a**) Initial cost volume C; (**b**) filtered cost volume C^A.

3.2. Cost Volume Collaborative Filtering

Aggregating the matching cost of pixels within each pixel's support window is a key step in eliminating matching ambiguity and suppressing noise. On the one hand, if only local support windows are used, then the matching ambiguity in largely weak-texture regions cannot be eliminated. On the other hand, if only tree structures from the whole image are used, then they are sensitive to high-texture regions. Therefore, aiming to cope with the matching ambiguity in different texture regions, dual support windows are constructed for each pixel p, namely a local adaptive support window as well as a global support window, which is the entire image as shown in Figure 4. On this basis, the cost volume is aggregated using an adaptive weighted guided filter with the cross-support window as the kernel window and a minimum spanning tree filter with the whole image window as the kernel window. By fusing the output from the two filters, we achieve the cost volume collaborative filtering strategy.

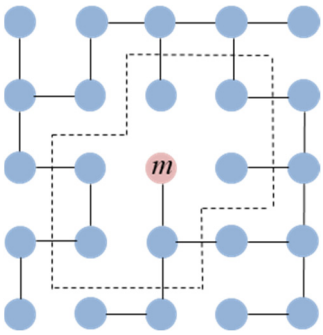


Figure 4. Local adaptive support window and global image window.

3.2.1. Cost Volume Filtering with Adaptive Weighted Guided Filter

A. The structure of cross-adaptive support window

For any pixel p, in order to obtain context information suitable for its structure, as shown in Figure 5, we construct four arms of left, right, up, and down on the reference images, and the horizontal arms of all pixels on its vertical arms form the cross-support window of the pixel p.

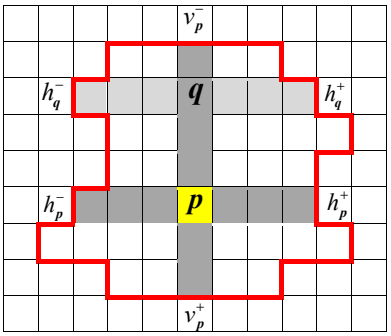


Figure 5. Four arms of pixel p and its cross-support window W_p.

We assume that the lengths of the arms are h_p^-, h_p^+, v_p^-, v_p^+, respectively; then, the end point and length of each arm can be determined according to the following judgment conditions:

① $L_{min} \le D_s(p_i, p) \le L_{max}$
② $D_c(\boldsymbol{p}_i, \boldsymbol{p}) < \tau \; and \; Mask(\boldsymbol{p}_i) = 0$
③ $D_c(\boldsymbol{p}_i, \boldsymbol{p}) < \frac{\tau}{2}, if \frac{L_{max}}{2} \le D_s(p_i, p) \le L_{max}$

In the above formulas, condition 1 defines the range of the arms, and L_{min} and L_{max} are the minimum arm length and the maximum arm length, respectively. $D_s(p_i,p) = |p_i - p|$

indicates the spatial distance between p_i and p, where p_i represents the i-th pixel in the horizontal or vertical direction from p. Condition 2 limits the color difference between adjacent pixels in the arm, $D_c(p_i, p) = \max_{c\in\{R,G,B\}} |I_c(p_i) - I_c(p)|$ represents the color distance between pixels p_i and p, and τ is the threshold of the color distance. In addition, $Mask(p_i) = 0$ limiting pixel p_i is not an edge point, and *mask* is the mask for *Canny* edge detection. Condition 3 indicates that when the arm length exceeds half of the maximum arm length, the threshold of the color distance is reduced by half for stricter inspection.

B. The cost volume is aggregated by an adaptive weighted guided filter

For any pixel p, it is assumed that its filtering result with the adaptive weighted guided filter on the d-th cost image is $C^A_{local}(p, d)$, and $C^A_{local}(p, d)$ is derived below according to the definition of the adaptive weighted guided filter.

It is assumed that W_k is an adaptive support window containing pixel p, and the filtering result of the adaptive weighted guided filter is the local linear transformation of the reference image I. Therefore, under the disparity level d, the filtering result $C^A_{local}(p, d)$ of $C(p, d)$ can be expressed as:

$$C^A_{local}(p, d) = a_k^T I(p) + b_k, \forall p \,\epsilon\, W_k \tag{4}$$

where a_k is a 3×1 coefficient vector, $I(p)$ is the 3×1 color vector of pixel p, and b_k is a scalar.

Estimating the parameters a_k and b_k is a linear regression problem. It can be solved by minimizing the sum of the absolute differences between the filtering inputs and outputs of all pixels in the window W_k. In order to better maintain disparity edges, inspired by weighted guided filtering in [33], this paper introduces a weighted factor $\psi(k)$ for perceiving edges into the regular terms of the objective function of linear regression, which is defined as follows:

$$\psi(k) = \frac{1}{N}\sum_{i=1}^{N} \frac{\sigma_I^2(k) + \lambda}{\sigma_I^2(i) + \lambda} \tag{5}$$

where $\sigma_I^2(i)$ is the gray variance of all pixels in the 3×3 window centered on pixel i after I is converted into a gray image, the constant λ is equal to $(0.001 \times M)^2$, M is the total number of gray levels, and N is the total number of all pixels.

Accordingly, the objective function $E(a_k, b_k)$ of the adaptive weighted guided filter is defined as follows:

$$E(a_k, b_k) = \sum_{p\in W_k} \left(\left(a_k^T I(p) + b_k - C(p, d) \right)^2 + \frac{\varepsilon}{\psi(k)} a_k^T a_k \right) \tag{6}$$

In the above equation, ε is the regularization parameter. The solutions of parameters a_k and b_k in the adaptive weighted guided filter are obtained by linear regression:

$$\begin{cases} a_k = \left(\Sigma_k + \frac{\varepsilon}{\psi(k)} J \right)^{-1} \left(\frac{1}{|W_k|} \sum_{p\in W_k} I(p) C(p, d) - \mu_k \overline{C(k, d)} \right) \\ b_k = \overline{C(k, d)} - a_k^T \mu_k \end{cases} \tag{7}$$

where Σ_k represents the color covariance matrix of all pixels in W_k, J is a 3×1 identity matrix, $|W_k|$ is the number of pixels in W_k, μ_k represents the color mean vector of all pixels in W_k, and $\overline{C(k, d)} = (\frac{1}{|W_k|}\sum_{p\in W_k} C(p, d))$ is the average matching cost of all pixels in W_k.

Finally, the filtering result $C^A_{local}(p, d)$ of pixel p in the d-th cost image is taken as the mean value of all pixels' filtering results in the support window W_p, that is:

$$C^A_{local}(p, d) = \frac{1}{|W_p|}\sum_{k\in W_p} \left(a_k^T I(p) + b_k \right) = \left(\frac{1}{|W_p|} \sum_{k\in W_p} a_k^T \right) I(p) + \frac{1}{|W_p|}\sum_{k\in W_p} b_k \tag{8}$$

3.2.2. Cost Volume Filtering with Minimum Spanning Tree Filter

The minimum spanning tree filter takes the entire reference image I as the kernel window. The reference image I is regarded as a connected and undirected graph, where each node corresponds to a pixel, and each edge connects a pair of adjacent pixels in the four-connected domain. The weight of each edge is the maximum of the RGB color differences between its adjacent pixels. As shown in Figure 4, it is easy to generate a minimum spanning tree from the undirected graph to connect all the pixels.

For any two pixels p and q, there is a unique path connecting them in the tree, and their distance $D(p,q)$ on the tree is equal to the sum of the weights of all the edges on their path. Then, the support weight $\omega(p,q)$ between p and q can be expressed as

$$\omega(p,q) = \exp\left(-\frac{D(p,q)}{\sigma}\right) \tag{9}$$

where the parameter σ is used to adjust the support weight between two pixels. The kernel function $k(p,q)$ of the minimum spanning tree filter can be expressed as

$$k(p,q) = \frac{\omega(p,q)}{\sum_{q\in I}\omega(p,q)} \tag{10}$$

It is worth noting that if pixels p and q are taken as two random variables, then $\omega(p,q)$ can be regarded as the joint probability of p and q, and the probability of p is equal to $\sum_{q\in I}\omega(p,q)$. According to Bayes' rule, the kernel function $k(p,q)$ represents that the conditional probability of q under p is known.

For any pixel p, let $C^A_{global}(p,d)$ denote its filtering result with the minimum spanning tree filter in the d-th cost image, which is the weighted sum of the matching costs in the whole image:

$$C^A_{global}(p,d) = \sum_{q\in I} k(p,q)C(q,d) = \frac{\sum_{q\in I}\omega(p,q)C(q,d)}{\sum_{q\in I}\omega(p,q)} \tag{11}$$

Here, we use the non-local cost aggregation method in [29] to quickly calculate the tree filtering results of all pixels at one time. It is assumed that $S^A(p,d)$ represents the molecule of $C^A_{global}(p,d)$ in Formula (11), i.e.,

$$S^A(p,d) = \sum_{q\in I}\omega(p,q)C(q,d) \tag{12}$$

$S^A(p,d)$ for each pixel can be calculated at once by traversing the minimum spanning tree twice in turn, as shown in Figure 6. In the first traversal, the cost aggregation is performed by traversing from leaves to roots as shown in Figure 6a, and the temporary aggregation cost $S^{A\uparrow}(p,d)$ of pixel p in this traversal can be calculated as follows:

$$S^{A\uparrow}(p,d) = \begin{cases} C(p,d) & if\ p\ is\ a\ leaf \\ C(p,d) + \sum_{q\in ch(p)}\omega(p,q)S^{A\uparrow}(q,d) & else \end{cases} \tag{13}$$

where $ch(p)$ is a set composed of all child nodes of pixel p.

In the second traversal, the cost aggregation is performed by traversing from the root to the leaves, as shown in Figure 6b. For the root node p, its final aggregation cost $S^A(p,d)$ can be calculated according to its father node $pr(p)$:

$$S^A(p,d) = \omega(pr(p),p)\cdot S^A(pr(p),d) + \left(1-\omega^2(pr(p),p)\right)\cdot S^{A\uparrow}(p,d) \tag{14}$$

Since the denominator of Formula (11) is a special case of the numerator with $C(q,d)\equiv 1$, it can be quickly solved by the above same method. Therefore, it is only necessary to traverse the tree four times to calculate the tree filtering result $C^A_{global}(p,d)$ of pixel p.

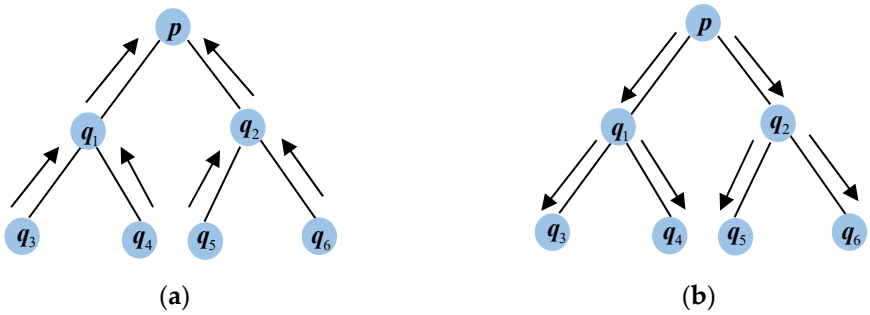


Figure 6. Two traversals on the tree. (**a**) From the leaves to the root; (**b**) from the root to the leaves.

3.2.3. Cost Volume Collaborative Filtering

For any pixel p, the collaborative filtering result $C^A(p,d)$ on the d-th cost image is equal to the average value of the filtering result $C^A_{local}(p,d)$ of the adaptive weighted guided filter and the filtering result $C^A_{global}(p,d)$ of the minimum spanning tree filter, namely:

$$C^A(p,d) = \frac{1}{2}\left(C^A_{local}(p,d) + C^A_{global}(p,d)\right) \tag{15}$$

3.3. Disparity Calculation and Disparity Refinement

In order to quickly calculate the disparity of each pixel, similar to other local optimization methods, this paper adopts the Winner-Takes-All (WTA) optimization method to obtain the optimal disparity of each pixel from the filtered cost volume $C^A(p,d)$. As shown in Figure 7, for pixel p, the WTA method directly selects the disparity d_p with its minimum filtering cost value from the disparity set $\mathcal{D}$, and then d_p is chosen as the optimal disparity of pixel p, i.e.,

$$d_p = \underset{d\in\mathcal{D}}{\operatorname{argmin}} C^A(p,d) \tag{16}$$

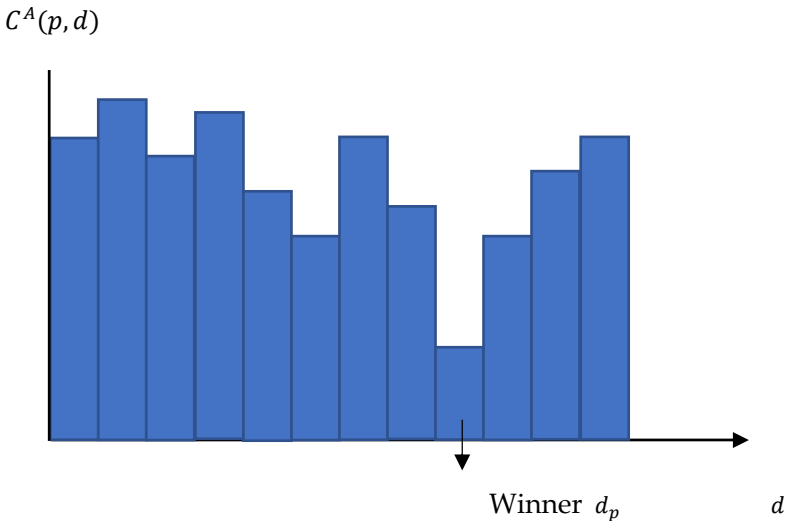


Figure 7. Winer-Takes-All optimization algorithm.

The raw disparity map can be generated after the optimal disparity of each pixel is calculated according to the WTA method. Considering that the raw disparity maps contain some noise points and outliers whose disparities are invalid, the method proposed in [34] is used to refine the disparity. Firstly, the left–right consistency check operation is used to detect the outliers. Secondly, the scanline filling method is used to fill a valid disparity for each outlier. Finally, a fast weighted median filter is used to smooth the filled disparity map, and then the final disparity map can be obtained.

4. Experimental Results and Analysis

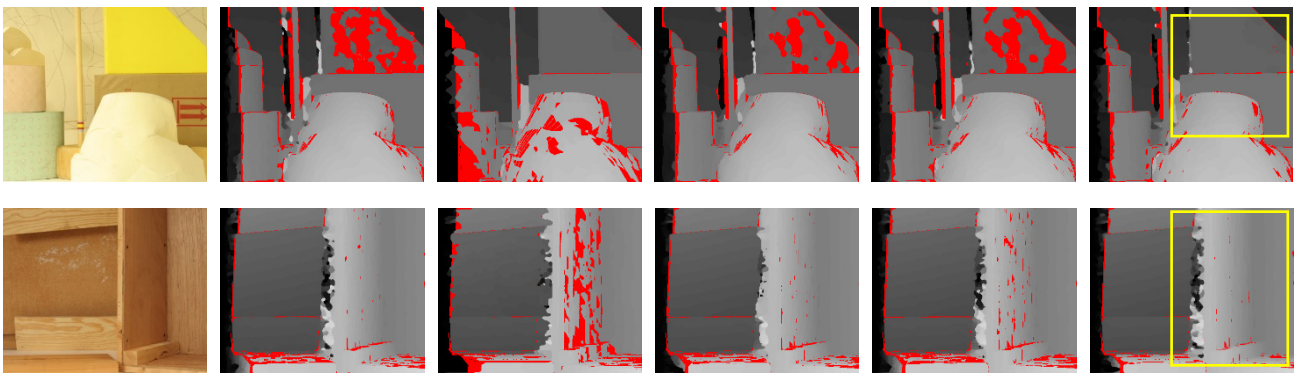
The stereo image dataset [10] provided by the Middlebury platform is used to verify the effectiveness of our proposed algorithm. The parameter settings of the proposed algorithm are shown in Table 1 and remain unchanged for all test images. The matching accuracy is measured by the error rate in the non-occluded regions, which is the percentage of mismatched pixels whose disparity error is greater than one pixel.

Table 1. The parameter settings of the proposed method.

Parameter	Value	Description
$\mathcal{N}_p$	5×7	The window size of the census transform
L_{min}	3	The minimum length of the arms of each pixel
L_{max}	15	The maximum length of the arms of each pixel
τ	6	The threshold of color distance for detecting each arm
ε	10^{-4}	The regularization parameter in the adaptive weighted guided filter
σ	0.03	The parameter for adjusting the support weight between two pixels

4.1. Evaluation of Cost Volume Filtering Method

In order to test the effectiveness of our cost volume filtering method, it is compared with the cost volume filtering methods in four state-of-the-art stereo matching algorithms (ISM [21], TF [28], CSGF [23] and GDGF [24]), and tested on 27 pairs of Middlebury stereo images in which objects contain various textures. First, the initial cost volume is generated with the matching cost calculation method in this paper, and it is used as the common input. Then, the above four cost volume filtering methods replace the proposed cost volume filtering method to generate their own raw disparity maps. Figure 8 shows the raw disparity maps of each cost volume filtering method, and the mismatched pixels are marked in red. Table 2 presents the error rates of different cost volume filtering methods on the raw disparity maps of each pair of stereo images.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 8. The raw disparity maps of Lampshade2 and Wood1 generated by different cost volume filtering methods. (**a**) Reference image; (**b**) ISM [21]; (**c**) TF [28]; (**d**) CSGF [23]; (**e**) GDGF [24]; (**f**) proposed.

It can be easily seen from Table 2 that our cost volume collaborative filtering method not only has the lowest average error rate, but also has the largest number of instances with the best results. Compared with the cost volume filtering methods of ISM, TF, CSGF, and GDGF, the average error rate of our cost volume cooperative filtering method is reduced by 1.55%, 3.41%, 1.14%, and 1.61%, respectively. From the quantitative results, the proposed cost volume collaborative filtering method is better than the other four cost volume filtering methods.

Table 2. Comparison of different cost volume filtering methods.

Data	ISM [21]	TF [28]	CSGF [23]	GDGF [24]	Proposed
Aloe	5.91	5.96	6.24	6.92	**4.14**
Art	12.13	12.41	11.60	12.01	**9.86**
Baby1	2.89	7.53	3.15	3.30	**2.31**
Baby2	3.68	14.04	3.79	**3.55**	3.81
Baby3	4.01	7.94	4.63	4.70	**4.09**
Books	9.86	11.05	9.53	9.74	**8.54**
Bowling1	6.02	16.97	5.67	5.88	**5.12**
Bowling2	6.98	12.22	7.84	7.18	**5.13**
Cloth1	1.02	0.98	1.68	1.79	**0.51**
Cloth2	2.85	4.50	3.76	4.12	**1.68**
Cloth3	2.28	2.51	2.44	2.76	**1.70**
Cloth4	1.89	1.94	2.04	2.44	**1.39**
Dolls	6.61	6.98	6.49	6.74	**4.54**
Flowerpots	9.45	16.22	9.34	9.33	**8.29**
Lampshade1	14.82	11.00	10.30	12.36	**7.08**
Lampshade2	16.84	12.66	11.01	14.01	**6.12**
Laundry	18.63	17.42	**16.38**	19.22	17.13
Moebius	10.10	10.76	10.20	9.69	**7.97**
Reindeer	6.78	11.22	7.16	7.60	**6.43**
Rocks1	3.25	3.52	3.43	3.44	**1.89**
Rocks2	2.26	2.87	2.40	2.53	**1.66**
Wood1	4.25	10.55	4.26	5.78	**2.72**
Wood2	**2.10**	5.73	2.53	2.32	2.43
Tsukuba	5.54	4.47	**3.66**	4.45	5.28
Venus	1.68	1.95	1.66	**1.49**	1.57
Teddy	8.43	**7.32**	7.91	7.97	7.50
Cones	4.25	4.00	4.27	4.63	**3.82**
Avg (%)	6.46	8.32	6.05	6.52	**4.91**

The best results are shown in bold among all methods.

According to the visual comparison results in Figure 8, ISM and GDGF adopt fixed support windows, which cause a lot of matching errors in largely weak-texture regions. CSGF decreases the matching ambiguity in the weak-texture regions by using a multi-scale cost aggregation model. TF proposes a fast non-local cost aggregation approach with a minimum spanning tree from the whole image, and it can eliminate the matching ambiguity in the weak-texture regions well, but it may produce matching errors easily in the high-texture regions, e.g., the side of Wood2 in Figure 8**.** In contrast, by constructing a local adaptive cross-support window and a global image support window, our method can not only maintain the inherent connection between neighborhood pixels, but also capture more effective context information, so that it can eliminate the matching ambiguity in different texture regions.

4.2. Evaluation of the Overall Performance of the Stereo Matching Algorithm

To prove the overall matching performance of our stereo matching method, the final disparity maps generated by our stereo matching method are evaluated on the Middlebury dataset, and they are compared with the final matching results of the above four stereo matching algorithms, i.e., ISM, TF, CSGF, and GDGF. For each pair of stereo images, Table 3 shows the error rates of its final disparity maps generated by different stereo matching algorithms. As indicated in Table 3, the average error rate from our stereo matching method is 3.17%, which is 2.19%, 3.93%, 1.59%, and 2.36% lower than that of the ISM, TF, CSGF, and GDGF methods, respectively. We also find from Table 3 that our stereo matching method achieves better results than the other four methods on most stereo images.

Table 3. Comparison of different stereo algorithms.

Data	ISM [21]	TF [28]	CSGF [23]	GDGF [24]	Proposed
Aloe	4.76	4.94	5.18	6.08	**2.93**
Art	9.34	10.41	8.83	9.47	**7.36**
Baby1	3.23	8.54	2.98	3.49	**1.57**
Baby2	2.33	15.49	**2.33**	2.50	2.96
Baby3	3.91	4.10	3.30	3.92	**2.05**
Books	8.44	9.60	7.55	8.96	**6.89**
Bowling1	11.68	20.80	9.30	10.95	**3.23**
Bowling2	4.88	11.06	4.74	5.43	**4.04**
Cloth1	0.44	0.48	0.84	0.93	**0.14**
Cloth2	2.18	3.97	2.96	3.32	**1.20**
Cloth3	1.68	1.91	1.90	2.10	**0.92**
Cloth4	1.37	1.23	1.50	1.63	**0.63**
Dolls	4.45	6.42	4.52	5.22	**3.46**
Flowerpots	9.79	15.26	8.28	9.73	**5.18**
Lampshade1	9.50	11.36	6.72	8.70	**2.79**
Lampshade2	19.17	10.71	16.36	17.42	**2.86**
Laundry	13.07	10.96	**10.31**	12.33	11.56
Moebius	9.58	7.97	8.29	8.70	**5.27**
Reindeer	3.81	8.57	4.28	5.49	**2.93**
Rocks1	2.93	2.70	2.71	2.96	**1.35**
Rocks2	1.27	2.07	1.35	1.50	**1.14**
Wood1	3.54	10.17	3.28	5.08	**1.37**
Wood2	0.34	1.47	**0.32**	**0.32**	0.37
Tsukuba	2.09	**1.52**	1.91	1.97	4.07
Venus	**0.17**	0.42	0.18	0.18	0.38
Teddy	7.37	6.34	6.04	7.52	**5.93**
Cones	3.34	3.22	**2.79**	3.27	3.09
Avg (%)	5.36	7.10	4.76	5.53	**3.17**

The best results are shown in bold among all methods.

Figure 9 shows the final disparity maps obtained by each matching method in which mismatched pixels are marked in red. We can see from the visual comparison in Figure 9, TF has the worst matching performance. This is mainly because TF destroys the original Markov structure of the reference image in the process of generating the tree, resulting in more mismatches in highly textured areas and depth discontinuities, e.g., the platform of Baby1 in Figure 5. Also, these disparity results cannot be improved even if they are refined by the post-processing operation. ISM, CSGF, and GDGF adopt local support windows and cannot effectively aggregate the matching cost within the context of each pixel, so they are unable to eliminate the matching ambiguity in largely weak-texture regions even after the disparity refinement operation, e.g., the yellow triangle area of Lampshade1 in Figure 9. We also find from the visual comparison that our stereo matching method adopts a local adaptive weighted guided filter and a global minimum spanning tree filter for the collaborative filtering of cost volume, which not only effectively eliminate the matching ambiguity in the weak-texture regions, but also maintain the disparity edge of each object well.

The final disparity maps obtained with our stereo matching method for six instances (i.e., Aloe, Cloth1, Bowling1, Rocks1, Moebius, and Reindeer) are shown in Figure 10. For each instance, from left to right are the reference images, the ground truth disparity maps, and the final disparity maps obtained with our method. As can be seen from Figure 10, the disparity maps obtained with our stereo matching method are very smooth and exceedingly close to the ground truth disparity maps. Accurate disparity results can be obtained in different texture regions, and the details such as edges in the disparity maps can also be well maintained.

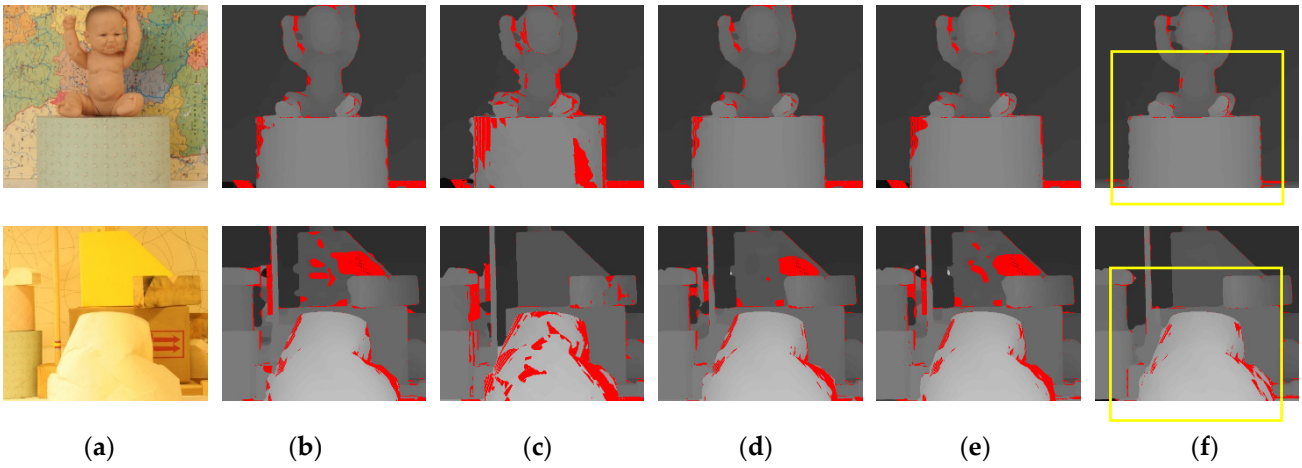


Figure 9. The final disparity maps of Baby1 and Lampshade1 generated by different stereo matching methods. (**a**) Reference image; (**b**) ISM [21]; (**c**) TF [28]; (**d**) CSGF [23]; (**e**) GDGF [24]; (**f**) proposed.

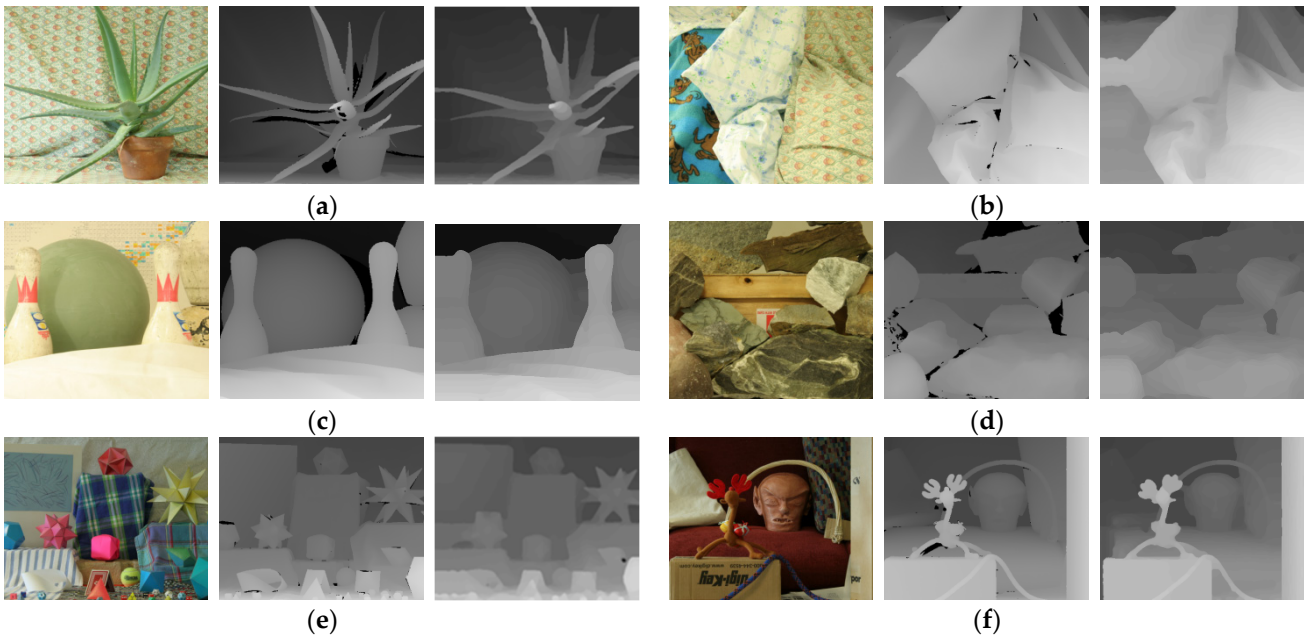


Figure 10. The ground truth disparity maps and the final disparity maps obtained with our method. (**a**) Aloe; (**b**) Cloth1; (**c**) Bowling1; (**d**) Rocks1; (**e**) Moebius; (**f**) Reindeer.

In order to compare the running time of the above five stereo matching methods in the experiment, all of them are implemented on the same PC, and their experiment environment configuration is also kept the same, as shown in Table 4. The average runtime of each stereo matching method on the above 27 pairs of Middlebury stereo images are shown in Table 5.

As shown in Table 5, ISM has the longest average runtime, and the reason is that ISM cascades an iterative guided filter and bilateral filter to filter the cost volume, which leads to exceedingly high computational complexity. TF calculates the aggregation cost of all pixels at the same time by traversing a tree twice, which results in extremely fast computational speed. CSGF needs to filter multiple cost volumes of different scales, so its running time is also long. The running time of our method is slightly longer than that of GDGF. This is because the cost volume needs to be collaboratively filtered by the local filter and the global filter in the proposed method, but GDGF uses a single down-sampling gradient domain guided filter to smooth the cost volume. However, our stereo matching method has higher

matching accuracy than the above four stereo matching methods and has stronger ability to eliminate the matching ambiguity.

Table 4. The experiment environment configuration.

CPU	Memory	Operating System	Programming Language
Intel(R) Core (TM) i7-10750	16 GB	Windows10 (64 bit)	C++ and OpenCV

Table 5. The runtime comparison of different stereo matching methods.

	ISM [21]	TF [28]	CSGF [23]	GDGF [24]	Proposed
Avg (s)	36.17	1.50	9.44	4.05	7.51

5. Conclusions

In this paper, we propose a novel stereo matching method with collaborative filtering of cost volume in order to effectively eliminate the matching ambiguity in the matching process. Firstly, for each pixel, a local cross-support window is constructed to maintain the original connectivity with its neighboring pixels in the local window. On this basis, an adaptive weighted guided filter with the cross-support window as the kernel window is derived to smooth the cost volume. Secondly, a minimum spanning tree is constructed in the whole image window to capture more effective context information in a global manner. Then, a minimum spanning tree filter is derived that can filter the cost volume quickly. By embedding the local adaptive weighted guided filter into the global minimum spanning tree filter, we can achieve the collaborative filtering of the cost volume. As such, each pixel can not only receive the correct support from the neighboring pixels in its local adaptive support window, but also obtain effective support from other pixels in the whole image. The evaluation results show that the average matching error rate of our method on the Middlebury stereo images is 3.17%. The experimental results demonstrate that the proposed stereo matching method with collaborative filtering of cost volume can not only eliminate the matching ambiguity of different texture regions, but also better preserve the disparity edge. It is noteworthy that the initial cost volume constructed by some matching cost calculation methods has a direct impact on the collaborative filtering of cost volume and the disparity accuracy. In future work, we will consider more outstanding and robust matching cost calculation methods that can be adapted to various severe conditions such as low resolution, image distortion, illumination changes, and so on, and then integrate them into our stereo matching framework in order to further improve the robustness of stereo matching. Moreover, the proposed stereo method will be accelerated by CUDA implementations on GPU systems to achieve real-time performance. As such, our stereo method will be applied to the stereo vision system of intelligent robots to restore the depth information of the scene and lay a solid foundation for subsequent 3D object detection or 3D reconstruction.

Author Contributions: Conceptualization, W.W. (Wenhuan Wu); investigation, W.W. (Wenhuan Wu); methodology, W.W. (Wenhuan Wu); software, X.X. and W.W. (Wenshu Wang); supervision, W.W. (Wenhuan Wu); validation, X.X.; visualization, H.Z.; writing—original draft preparation, W.W. (Wenhuan Wu) and X.X.; writing—review and editing, W.W. (Wenshu Wang) and H.Z.; funding acquisition, W.W. (Wenhuan Wu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Natural Science Foundation of Hubei Province (NO.2022CFB538), China; by Theme Case Project of the Degree and Graduate Education Development Center of the Ministry of Education (NO. ZT-231052501), China; by Ph.D. Research Startup Foundation Project of Hubei University of Automotive Technology (NO. BK202347), China.

Data Availability Statement: The datasets used in this study were sourced from Middlebury Stereo Datasets at https://vision.middlebury.edu/stereo/data/ (accessed on 5 February 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shi, Y.; Guo, Y.; Mi, Z.; Li, X. Stereo CenterNet-based 3D object detection for autonomous driving. *Neurocomputing* **2022**, *471*, 219–229. [CrossRef]
2. Cui, J.; Min, C.; Feng, D. Research on pose estimation for stereo vision measurement system by an improved method: Uncertainty weighted stereopsis pose solution method based on projection vector. *Opt. Express* **2020**, *28*, 5470–5491. [CrossRef] [PubMed]
3. Tian, X.; Liu, R.; Wang, Z.; Ma, J. High quality 3D reconstruction based on fusion of polarization imaging and binocular stereo vision. *Inf. Fusion* **2022**, *77*, 19–28. [CrossRef]
4. Li, Y.; Zheng, S.; Wang, X.; Ma, H. An efficient photogrammetric stereo matching method for high-resolution images. *Comput. Geosci.* **2016**, *97*, 58–66. [CrossRef]
5. Taniai, T.; Matsushita, Y.; Sato, Y.; Naemura, T. Continuous 3D label stereo matching using local expansion moves. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 2725–2739. [CrossRef] [PubMed]
6. Xu, Y.; Yu, D.; Ma, Y.; Li, Q.; Zhou, Y. Underwater stereo-matching algorithm based on belief propagation. *Signal Image Video Process.* **2022**, *17*, 891–897. [CrossRef]
7. Yao, P.; Zhang, H.; Xue, Y.; Chen, S. As-global-as-possible stereo matching with adaptive smoothness prior. *IET Image Process.* **2019**, *13*, 98–107. [CrossRef]
8. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [CrossRef]
9. Laga, H.; Jospin, L.V.; Boussaid, F.; Bennamoun, M. A survey on deep learning techniques for stereo-based depth estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1738–1764. [CrossRef]
10. Scharstein, D.; Szeliski, R. Middlebury Stereo Datasets. [Online]. Available online: https://vision.middlebury.edu/stereo/data/ (accessed on 5 February 2024).
11. Yang, S.; Lei, X.; Liu, Z.; Sui, G. An efficient local stereo matching method based on an adaptive exponentially weighted moving average filter in SLIC space. *IET Image Process.* **2021**, *15*, 1722–1732. [CrossRef]
12. Tatar, N.; Arefi, H.; Hahn, M. High-resolution satellite stereo matching by object-based semiglobal matching and iterative guided edge-preserving filter. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1841–1845. [CrossRef]
13. Zhang, K.; Lu, J.; Lafruit, G. Cross-based local stereo matching using orthogonal integral images. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 1073–1079. [CrossRef]
14. Yoon, K.J.; Kweon, I.S. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 650–656. [CrossRef] [PubMed]
15. Richardt, C.; Orr, D.; Davies, I.; Criminisi, A.; Dodgson, N.A. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In Proceedings of the 11th European Conference on Computer Vision, Hersonissos, Greece, 5–11 September 2010; pp. 510–523.
16. Yang, Q. Hardware-efficient bilateral filtering for stereo matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1026–1032. [CrossRef] [PubMed]
17. He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. [CrossRef] [PubMed]
18. Hosni, A.; Rhemann, C.; Bleyer, M.; Rother, C.; Gelautz, M. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 504–511. [CrossRef] [PubMed]
19. Yang, Q.; Ji, P.; Li, D.; Yao, S.; Zhang, M. Fast stereo matching using adaptive guided filtering. *Image Vis. Comput.* **2014**, *32*, 202–211. [CrossRef]
20. Lu, J.; Shi, K.; Min, D.; Lin, L.; Do, M.N. Cross-based local multipoint filtering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 430–437.
21. Hamzah, R.A.; Kadmin, A.F.; Hamid, M.S.; Ghani, S.F.A.; Ibrahim, H. Improvement of stereo matching algorithm for 3D surface reconstruction. *Signal Process. Image Commun.* **2018**, *65*, 165–172. [CrossRef]
22. Hamzah, R.A.; Ibrahim, H.; Hassan, A.H.A. Stereo matching algorithm based on per pixel difference adjustment, iterative guided filter and graph segmentation. *J. Vis. Commun. Image Represent.* **2017**, *42*, 145–160. [CrossRef]
23. Zhang, K.; Fang, Y.; Min, D.; Sun, L.; Yang, S.; Yan, S.; Tian, Q. Cross-Scale Cost Aggregation for Stereo Matching. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 965–976. [CrossRef]
24. Yuan, W.; Meng, C.; Tong, X.; Li, Z. Efficient local stereo matching algorithm based on fast gradient domain guided image filtering. *Signal Process. Image Commun.* **2021**, *95*, 116280. [CrossRef]
25. Fan, S.; Sun, W.; Zheng, J.; Fu, Q.; Xue, M.; Wu, W. Accurate edge-preserving stereo matching by enhancing anisotropy. *Signal Process. Image Commun.* **2023**, *114*, 116945. [CrossRef]
26. Cigla, C. Recursive edge-aware filters for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 27–34.
27. Pham, C.C.; Jeon, J.W. Domain transformation-based efficient cost aggregation for local stereo matching. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *23*, 1119–1130. [CrossRef]
28. Yang, Q. Stereo matching using tree filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 834–846. [CrossRef] [PubMed]

29. Yang, Q. A non-local cost aggregation method for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1402–1409.
30. Mei, X.; Sun, X.; Dong, W.; Wang, H.; Zhang, X. Segment-tree based cost aggregation for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 313–320.
31. Cheng, F.; Zhang, H.; Sun, M.; Yuan, D. Cross-trees, edge and superpixel priors-based cost aggregation for stereo matching. *Pattern Recognit.* **2015**, *48*, 2269–2278. [CrossRef]
32. Hirschmuller, H.; Scharstein, D. Evaluation of cost functions for stereo matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
33. Li, Z.; Zheng, J.; Zhu, Z.; Yao, W.; Wu, S. Weighted guided image filtering. *IEEE Trans. Image Process.* **2015**, *24*, 120–129.
34. Wu, W.; Zhu, H.; Yu, S.; Shi, J. Stereo matching with fusing adaptive support weights. *IEEE Access* **2019**, *7*, 61960–61974. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.