

A PROGRESSIVE STEREO MATCHING ALGORITHM BASED ON TEXTURE EXTENSION

Yingjiang Li¹, Yuzhong Zhong² and Maoning Wang²

¹Chongqing university of technology, Chongqing, China 400054

²Sichuan University, Chengdu, Sichuan, China 610065

ABSTRACT

It is difficult to get accurate stereo matching for the occluded and smooth areas of an image. In this paper, a stereo matching algorithm is proposed based initially on texture region and then on smooth region. First, the reasonable texture points in the left and right image pairs are obtained, and the texture point disparity values of the left and right images are calculated respectively using the adaptive-weight cost aggregation method. Then, starting from the texture points, the disparity values of the smooth areas are calculated by expanding the disparity maps of the left and right textures. Finally, the expanded left and right disparity maps are fused to obtain the final one. In the matching process, the algorithm effectively avoids the occluded area and adds constraint methods to improve the matching accuracy and speed. Experiments show that the proposed algorithm is fast and has higher accuracy disparity between discontinuous and smooth regions.

KEYWORDS

stereo matching, adaptive weights, texture extraction, disparity expansion

1. INTRODUCTION

Binocular stereo vision is an important branch of computer vision and is widely used in the fields of 3D reconstruction, robot and unmanned aerial vehicle (UAV) navigation, and automatic driving. Among these, stereo matching, as a key concept in stereo vision, has become a hot topic in the field of computer vision research [1] and the method for acquiring accurate disparity images quickly and efficiently is of particular significance.

At present, region-based stereo matching methods can be divided into local, semi-global, and global stereo matching. Global stereo matching establishes a minimum energy function for one line or all areas of an image to obtain the optimal disparity values, such as graph cut [2], confidence propagation [3], and dynamic programming [4]. The global algorithm can generate a more accurate disparity map, but it is difficult to achieve fast stereo matching due to high computational complexity. To improve the matching speed, Hirschmuller proposed a semi-global

Natarajan Meghanathan et al. (Eds) : ICCSEA, WiMoA, SCAI, SPPR, InWeS, NECO - 2019

pp. 111-121, 2019. © CS & IT-CSCP 2019

DOI: 10.5121/csit.2019.91809

algorithm [5] reducing the optimization problem of the global algorithm from two dimensions to one. This is currently one of the most successful algorithms.

Adaptive weight [6] is a good local stereo matching algorithm. This method determines the weight of pixels in the window according to the similarity of colors and geometric distance between pixels and improves the matching accuracy of disparity discontinuous regions. The cost aggregation method based on bilateral filtering proposed by Mattoccia et al [7] is similar to the adaptive weight method. The image segmentation method [8] uses the segmentation results to adaptively select the size of the matching window and improve the matching accuracy of disparity discontinuity regions, but the image segmentation process affects the efficiency of the whole algorithm. Drouyer [9] used a waterfall hierarchical segmentation and robust regression model to propagate disparity values, improving the efficiency of the segmentation algorithm. To improve the robustness of the algorithm in dealing with noise and image brightness changes, some algorithms use Rank transform [10] or census transform [11] to calculate the matching cost. Another kind of algorithm considers the texture information [12, 13] of an image and uses the corresponding relationship of the edge points between the current and the previous frames to quickly obtain the sparse disparity of the video. The method [14] uses the non-texture region constraint and edge region constraint to improve the accuracy of stereo matching.

Another kind of matching algorithm uses a deep learning method to train the set of image pairs and disparity maps to obtain depth information, and develops very accurate disparity maps, such as [15-17]. However, this kind of algorithm requires expensive equipment and long training times. Also, the disparity result is strongly dependent on the image set used in the training process.

For now, obtaining accurate disparity estimation in disparity discontinuous and weak texture regions remains a challenge. Therefore, we propose a progressive stereo matching method. First, the texture region of the image is extracted, and the texture disparity map of the left and right images is obtained by using the adaptive-weight cost aggregation method. Then the left texture disparity map is expanded to the right, and the right texture disparity map to the left to obtain disparity maps of the left and right images, respectively. Finally, the left and right disparity maps are fused to obtain the final one. This disparity estimation method can effectively reduce error matching in occlusion and smooth regions.

2. PROPOSED STEREO MATCHING METHOD

The method proposed in this paper is mainly divided into two steps: texture point acquisition and disparity estimation; expansion and fusion of texture disparity map. In this section, we describe these two steps in detail.

2.1. Texture Point Acquisition and Disparity Estimation

The image features of texture regions are obvious, and accurate disparity is easier to obtain when matching. Here we need to obtain reasonable texture points and an accurate texture disparity map.

2.1.1. Texture Point Acquisition

The process for obtaining image texture is similar to image edge detection. There are many methods including differential operator, Laplace of Gaussian (Log) operator, and canny operator.

However, these methods do not meet our requirements. Therefore, we propose a different method to obtain the texture of an image. Whether a point is a texture point can be identified by Formula (1)

$$T_p = \begin{cases} 255 & \text{if } \left(\text{count}_{q \in P} (|I_p - I_q| > \omega) \right) > 5 \\ 0 & \text{else} \end{cases} \quad (1)$$

Where p is a pixel in the image; T_p is the value of the texture matrix; and q is the pixel in the neighborhood window (3×3) of point p . I_p, I_q are gray values of point p and q ; ω is the differential threshold. Using formula (1), the number of gray scale differences greater than ω between one pixel and adjacent pixels is counted. If the number exceeds 5, pixel points are considered as texture points; otherwise they are considered smooth points. Taking the Teddy image as an example, the texture extraction results are shown in Figure 1.



Figure 1. Texture image obtained by the proposed method. Left is the gray image of Teddy, right is the texture image (where the value of 0 represents smooth point and 255 represents texture point).

2.1.2. Texture Disparity Estimation

In this paper, we need to estimate the texture disparity of the left and right image respectively. The adaptive weights matching cost aggregation method is used in the estimation process. The traditional adaptive weight stereo matching algorithm was proposed by Yoon [6], and the support weight function of its neighboring pixels can be expressed as:

$$w(p, q) = f_c(\Delta c_{pq}) \cdot f_d(\Delta d_{pq}) \quad (2)$$

Where p is the pixel to be matched; q is the pixel point in the matching window; and $w(p,q)$ is the weight of the corresponding point in the window; f_c and f_d represent the color similarity and spatial proximity of points p and q , respectively. Δc_{pq} and Δd_{pq} can be expressed as:

$$\Delta c_{pq} = \sqrt{(r_p - r_q)^2 + (g_p - g_q)^2 + (b_p - b_q)^2} \quad (3)$$

$$\Delta d_{pq} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \quad (4)$$

Where (r_p, g_p, b_p) is the color component of point p in RGB color space; (x_p, y_p) is the coordinate position of point p .

To improve the matching accuracy of texture points, gradient similarity is added to the weight calculation. Similar to Laplacian operator, the gradient calculation of point p is as follows:

$$g_p = I_p(x+1, y) + I_p(x-1, y) + I_p(x, y+1) + I_p(x, y-1) - 4I_p(x, y) \quad (5)$$

Then the gradient similarity of point p and point q is expressed as:

$$\Delta g_{pq} = |g_p - g_q| \quad (6)$$

The weight function of neighboring pixels can be rewritten as:

$$w(p, q) = f_c(\Delta c_{pq}) \cdot f_d(\Delta d_{pq}) \cdot f_g(\Delta g_{pq}) \quad (7)$$

Where f_c , f_d and f_g use the Gaussian kernel function to calculate the corresponding similarity and proximity:

$$f_c = f_d = f_g = \exp\left(-\frac{x}{\delta}\right) \quad (8)$$

Finally, the form of the adaptive weight function can be expressed as:

$$w(p, q) = \exp\left(-\frac{\Delta c_{pq}}{\delta_c}\right) \cdot \exp\left(-\frac{\Delta d_{pq}}{\delta_d}\right) \cdot \exp\left(-\frac{\Delta g_{pq}}{\delta_g}\right) \quad (9)$$

Where $\delta_c, \delta_d, \delta_g$ are adjustable parameters. The matching cost aggregation process is the same as that in reference [6]. In order to improve the matching speed and accuracy, we propose some constraints:

- (1) Corresponding matching points in left and right images should have similar gray or intensity values.
- (2) The corresponding matching point \bar{p} on the right image should also be a texture point or a neighboring texture point.

- (3) In order to reduce the mismatching of occluded areas, the matched points on the right image are marked.
- (4) For points on the same texture, the disparity value of the current point restricts the disparity search range of subsequent ones.

Through the above constraint strategy, the matching speed and mismatching rate are both improved. After left and right consistency (LRC) check, the matching results are shown in Figure 2. Here we use the pseudo-color to display disparity maps.



Figure 2. Left and right texture disparity map

2.2. Expansion and Fusion of the Texture Disparity Map

In this section we estimate the disparity value of the smooth region according to the texture disparity value. For image pairs acquired by cameras placed in parallel to left and right, if the texture itself in the left image is not occluded in the right image, then the right side of the texture must not be occluded in the right image; similarly, if the texture itself in the right image is not occluded in the left image, the left side of the texture must not be occluded in the left image.

According to this characteristic, for the left image, we estimate the disparity value of the smooth region from the texture point to the right. For the right image, we estimate the disparity value of the smooth region from the texture point to the left. This ensures that no points will be occluded in the matching process, i.e., corresponding matching points must be found in another reference image.

2.2.1. Texture Disparity Map Extension

When expanding disparity from texture to smooth area, we regard each smooth area segment in horizontal direction as a whole; that is, each smooth segment is also smooth in disparity value, and there will be no sudden change in disparity. Setting the overall disparity of the smooth segment as D_s , if the length S_L of the smooth segment is smaller than the diameter w_{2r} of the

matching cost aggregation window, then the disparity of all pixels of the smooth segment is calculated in pixels. If the smooth segment is longer than w_{2r} , the overall disparity value D_s of the segment is calculated in segments, and then the disparity of each pixel in the segment is calculated with D_s as the constraint.

For the left image, the disparity calculation process of the smoothed segment is as follows:

A. If $S_L \leq w_{2r}$, disparity estimation is performed in pixels, and the matching cost aggregation method is the same as the texture point matching method.

B. If $S_L > w_{2r}$, estimate the global disparity of the segment first, and then estimate the disparity of each pixel of the segment.

B1. Find a matching segment in the right image at the cost of matching all pixels contained in the segment. The right side of the segment may be occluded and the high matching cost area on the right side of the segment should be ignored when matching. The segment matching cost formula is as follows:

$$E(s_l, s_r) = \frac{\sum_{p \in s_l, q \in s_r, |I_p - I_q| \leq \sigma_s} |I_p - I_q|}{S_L'} \quad (10)$$

where s_l, s_r are the corresponding matching segments in the left and right images, respectively; p is the pixel within s_l and q is the pixel within s_r ; I_p, I_q represent the intensity values of points p and q , respectively; S_L' is the number of pixels in the segment that participate in calculating the matching cost; σ_s is a high matching cost threshold. The segment with the lowest matching cost is taken as the final matching result, and the value obtained is taken as the disparity value D_{sL} of the whole segment. Since the left side of the smooth segment is not occluded, the disparity search range is $D_{sL} < d_{t1} + S_L/2$ when segment matching is performed, where d_{t1} is the disparity value of the left texture point.

B2. The smooth region may be an inclined plane, and so there will be some differences between the disparity value within the smooth region and D_{sL} . Since the disparity value of the left end of the segment will not be greater than the disparity value of its left texture point, the disparity range at the left end of the segment is $[D_{sL} - S_L/2, d_{t1}]$. Perform stereo matching to the left end of segment with this range constraint, and the subsequent pixel points of the segment are matched with smooth constraint, i.e., $|d_i - d_{i-1}| \leq 2$, where d_i and d_{i-1} are the disparity values of the current and the previous pixels in the segment, respectively. Each pixel also uses the method of adaptive weight for cost aggregation when matching.

For the right image, we can use the same method to get the disparity value of the smooth region, but in the expansion process, we expand from the texture point to the left side according to the right texture map.

2.2.2. Fusion of Disparity Maps

After expanding the texture disparity map of the left and right images, we have two disparity maps, the left disparity map $disp_L$ and the right disparity map $disp_R$. There will be no disparity value in some areas of $disp_L$ and $disp_R$. By fusing the two disparity images, a complete disparity map can be obtained. At the same time, LRC detection can eliminate mismatched disparity values

in the fusion process. With the left disparity map as the final disparity image, the fusion process is as follows:

- A. If a pixel point has no value in the left disparity map but has a value in the corresponding point of the right disparity map, the pixel is assigned the disparity value of the right map.
- B. If a pixel has a disparity value in both the left and right disparity maps, assess the difference between the two disparity values. If $|d_{pl} - d_{pr}| \leq 2$ (d_{pl} and d_{pr} are the disparity values of point p in the left and right disparity maps respectively), then take d_{pl} as the disparity value of point p ; if $|d_{pl} - d_{pr}| > 2$, the disparity value at that point is deleted.

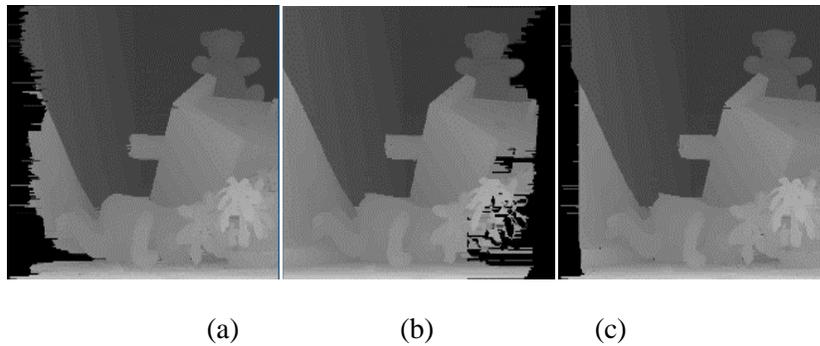


Figure 3. Disparity expansion and fusion results. (a) is the left expansion disparity map, (b) is the right expansion disparity map, and (c) is the fusion disparity map.

After the fusion is completed, it is necessary to interpolate and fill in the empty disparity areas caused by mismatching and occlusion in the disparity map, using the interpolation method of similar neighboring pixels as the filling method. Finally, the refinement operation method in [18] is used to refine the whole disparity map in sub-pixels. The whole stereo matching process is complete, and the experimental results of the texture disparity expansion and the fusion process are shown in Figure 3.

3. EXPERIMENT

In the performance evaluation of the algorithm, this paper uses the dataset provided by the Middlebury website [19]. In comparing this process with other algorithms and in order to accurately evaluate the matching performance of the algorithm itself, no disparity refinement method was added to the experimental results. All the algorithms in the experiment use the same PC platform configured with 3.40 GHz Intel Core i7 CPU, 8G RAM, and 64-bit OS. Also, all algorithms run under C++ code.

We selected existing classical algorithms such as SGM (Semi-Global matching) [5] and CS (Cross-Scale) [20] for our experimental comparison. For the experiment, some parameters in this paper were set as follows: difference threshold $\omega = 5$; adaptive weight adjustment parameters

$\delta_c = 10, \delta_d = 20, \delta_g = 5$; the diameter of the cost aggregation window $w_{2r} = 15$; threshold for high matching cost $\sigma_s = 10$. In addition, for the CS method, the number of iterations in the CS algorithm was 3, the matching window size was 25 pixels, and the parameters of other algorithms remained at their original default values. The disparity map of the experimental results is shown in Figure 4.

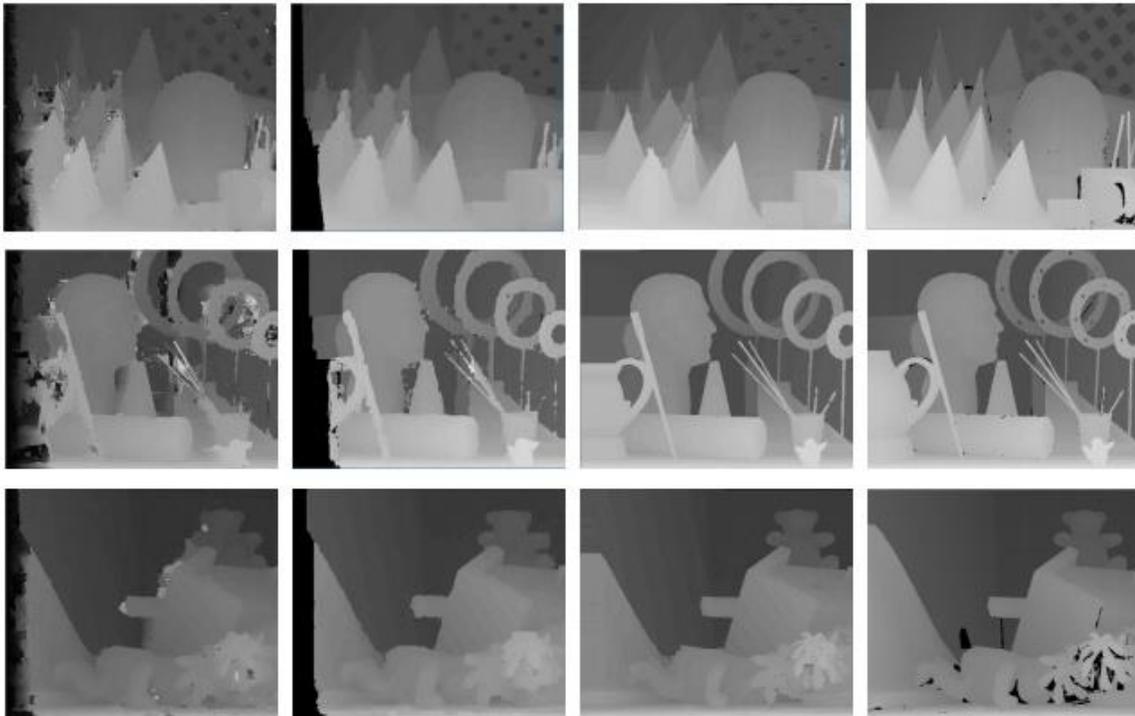


Figure 4. Unrefined disparity map. From top to bottom: Cone, Art, Teddy; From left to right: CS algorithm, SGM algorithm, this algorithm and ground truth.

In addition, this paper also uses standard evaluation bad 2.0 to compare the disparity accuracy of these algorithms. The error percentage is compared according to the occlusion area, all areas, and discontinuous areas respectively. The results are shown in Table 1.

As can be seen from Figure 4 and Table 1, the error matching rate of the algorithm proposed in this paper is the lowest in the non-occlusion area and all areas. This is because the algorithm performs texture matching first and then expands to smooth region, avoiding false matching of occluded regions and improving the matching accuracy of images for disparity discontinuous regions. In addition, we performed stereo matching for all images in the training set of the 2014 dataset provided by Middlebury. The average running time of each algorithm was: SGM, 8.6 seconds; CS, 130.4 seconds; our method, 8.7 seconds. The algorithm proposed in this paper also has a fast speed.

Table 1. Error Percentage Evaluation Results of Stereo Matching Algorithms (Error Threshold = 2.0)

Algorithm		SGM	CS	Our method
Art	Nonocc	3.03	4.75	2.87
	all	14.47	22.80	13.36
	disc	6.25	11.86	6.10
Teddy	Nonocc	4.08	5.91	3.21
	all	12.16	19.64	9.22
	disc	13.77	16.84	11.58
Cones	Nonocc	2.14	2.81	1.90
	all	6.42	8.73	6.65
	disc	5.50	9.32	4.73
Average Error(Nonocc)		3.08	4.49	2.66

Fig. 5 shows the performance of the change of three parameters δ_c , δ_d and δ_g on the error of non-occluded areas. The error percentage in the Fig.5 is the average value calculated from 15 pairs of images in the training data set of Middlebury. In each case, we keep the other parameters constant. Each parameter change has little influence on the matching accuracy of the proposed algorithm because the effect of outliers is suppressed by using the combined form of support-weights.

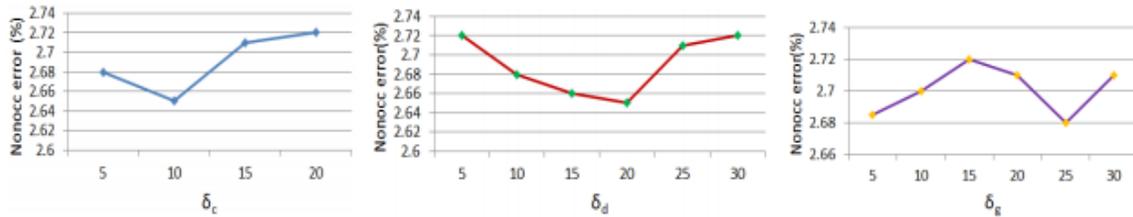


Figure 5. The performance of the proposed algorithm when three parameters are changed separately.

4. CONCLUSION

In order to improve the false matching caused by occlusion and weak texture regions in images, this paper proposes a progressive stereo matching method: texture regions first and smooth regions later. The algorithm first extracts the texture region of the image, calculates the disparity value of the two texture images by using the adaptive weight method, and performs an LRC check. Then, according to the texture disparity maps, the left image expands to the right and the right image to the left, and the respective disparity maps of the smooth regions are calculated. The left and right disparity maps are then fused and the final disparity map is obtained through disparity refinement. Experiments show that the algorithm proposed in this paper is fast and improved accurate disparity estimation can be obtained in both smooth and disparity discontinuity regions of the image.

ACKNOWLEDGEMENT

The work was supported by Scientific Research Foundation of Chongqing University of Technology, China(Grant No.2019ZD112) and Research Foundation of science and technology department of Sichuan, China (Grant No. 2018RZ0080).

REFERENCE:

- [1] Bleyer, Michael, and C. Breiteneder. Stereo Matching State-of-the-Art and Research Challenges. *Advanced Topics in Computer Vision*. Springer London, 2013:143-179.
- [2] Hong, Li, and G. Chen. "Segment-based stereo matching using graph cuts." *Computer Vision and Pattern Recognition, 2004.CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on IEEE*, 2004:I-74-I-81 Vol.1.
- [3] Sun, Jian, H. Y. Shum, and N. N. Zheng. "Stereo Matching Using Belief Propagation." *European Conference on Computer Vision Springer-Verlag*, 2003:510-524.
- [4] Kim, Jae Chul, et al. "A dense stereo matching using two-pass dynamic programming with generalized ground control points." *Computer Vision and Pattern Recognition, 2005.CVPR 2005. IEEE Computer Society Conference on IEEE*, 2005:1075-1082.
- [5] Hirschmuller, Heiko. "Stereo Processing by Semiglobal Matching and Mutual Information." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 30.2(2007): 328-341.
- [6] Yoon, Kuk Jin, and I. S. Kweon. "Adaptive Support-Weight Approach for Correspondence Search." *IEEE Trans Pattern Anal Mach Intell* 28.4(2006):650-656.
- [7] Mattoccia, Stefano, S. Giardino, and A. Gambini. "Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering." *Asian Conference on Computer Vision Springer-Verlag*, 2012:371-380.
- [8] Bleyer, M., and M. Gelautz. "A layered stereo algorithm using image segmentation and global visibility constraints." *Isprs Journal of Photogrammetry & Remote Sensing* 59.3 (2005):128-150.
- [9] Drouyer S, Beucher S, Bilodeau M, et al. Sparse Stereo Disparity Map Densification Using Hierarchical Image Segmentation[J]. *Springer*, 2017:172-184
- [10] Guo, Long Yuan, Y. Q. Xia, and J. Y. Yang. "Research and Application of Stereo Matching Based on Rank Transform." *Journal of System Simulation* 19.9(2007):2121-2117.
- [11] Bae, KyeongRyeol, et al. "A census-based stereo matching algorithm with multiple sparse windows." *International Conference on Ubiquitous & Future Networks IEEE*, 2015:240-245.
- [12] Jaafari, Ilyas El, et al. "Fast spatio-temporal stereo matching for advanced driver assistance systems." *Neurocomputing* 194.C(2016):24-33.
- [13] Koutti, Lahcen, I. E. Jaafari, and M. E. Ansari. "Temporal consistent stereo matching approach for road applications." *Computer Systems and Applications IEEE*, 2017.

- [14] Kim, Kyung Rae, and C. S. Kim. "Adaptive smoothness constraints for efficient stereo matching using texture and edge information." IEEE International Conference on Image Processing IEEE, 2016:3429-3433.
- [15] Zbontar, Jure, and Y. Lecun. "Computing the stereo matching cost with a convolutional neural network." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 2015:1592-1599.
- [16] Lecun, Yann. Stereo matching by training a convolutional neural network to compare image patches. JMLR.org, 2016.
- [17] Bailer, Christian, K. Varanasi, and D. Stricker. "CNN-Based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss." (2016).
- [18] Yang, Qingxiong, et al. "Spatial-Depth Super Resolution for Range Images." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 2007:1-8.
- [19] Scharstein, Daniel, et al. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. Pattern Recognition. Springer International Publishing, 2014:31-42.
- [20] Zhang, Kang, et al. "Cross-Scale Cost Aggregation for Stereo Matching." Computer Vision and Pattern Recognition IEEE, 2014:1590-1597.

AUTHORS

Yingjiang Li received Ph.D. degree from the School of Computer Science of Sichuan University, Sichuan, China, in 2019 ; and received the B.E. degree from the Department of Computer Science and Technology, Yunnan Normal University, Yunnan, China, in 2007. He has authored many papers on image denoizing and stereo matching in the journal of the relevant field. His current research interests include stereo matching and image enhancement.



Yuzhong Zhong received Ph.D. degree from the School of Computer Science of Sichuan University, Sichuan, China, in 2019 ; and received the M.S. degree from the school of Electronic Information Engineering, Civil Aviation University of China, Tianjin, China, in 2012. She is studying for a postdoctoral degree in the School of Communication of Sichuan University. Her research interests include machine learning and computer vision.



Maoning Wang received the Master degree of Electrical & Electronic in Sichuan University, Sichuan, China, in 1996, as well as M.S. of Computer Science in Carleton University, Ottawa, Canada, in 2003. Currently, she is pursuing the Ph.D. degree with Department of Computer Science and Technology, Sichuan University, Sichuan. She is also an associate professor in Sichuan University. Her research focuses on image processing, AI and sensor networks.

