International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.6 (2016), pp.35-46
http://dx.doi.org/10.14257/ijsip.2016.9.6.04

Adaptive Weight Stereo Matching Algorithm based on Neural Network and Improved Mean-Shift

Zheng Sun [1], Rong Yang [2] and Hui Li [3]

[1] School of Mechatronic Engineering, Zaozhuang University, Zaozhuang 277160,
[2] Department of Mechanical Engineering, McMaster University, ON L8S 1A1,
[3] School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 221116,
[1]China; cumt_sz@163.com;[2]Canada; yangr27@mcmaster.ca; [3]China; lihui09050013@gmail.com.

Abstract

In order to increase the effect of matching for local stereo matching method and decrease the amount of computation, a new adaptive weight based local stereo matching method is proposed in this paper. In this method, two methods are mainly employed to construct weight model: (1) Neural Network is used to establish the spatial weight model, which makes good use of the pixels in support window; (2) An edge hold off Mean-Shift method is proposed to distribute the intensity weight accurately. For decreasing the matching cost error, the census transform is introduced to calculate the matching cost. The influence of the parameters on the performance of our method is also discussed at last. Simulation results indicate that the performance of our method is better than that of Yoon's method under low support window.

***Keywords**: Neural Network, Edge Hold Off, Stereo Vision, Adaptive Weight*

1. Introduction

The stereo matching is widely used in 3D reconstruction, robot stereo vision navigation. According to the difference on optimization method, the stereo matching can be divided into many classes, mainly including the local methods and the global methods [1].

The global matching mainly includes the dynamic programming [2-5], the graph cut method [6-9], the belief propagation [10-13] and the partial differential equation method [14-16]. This kind of method obtains the disparity map through minimizing the energy formula (1). The global matching can get more accurate disparity map, whose disadvantage is longer operation time.

$$E(f) = D_{Data}(f) + D_{Smooth}(f) \tag{1}$$

The local matching is a common way of stereo matching, which has simpler structure, faster matching speed, and wider application. The choice of window size is the key point to improve the accuracy of local matching. On the other hand, the window size should be as big as possible to cover the gray variation well for reliable matching. If not, it could cause disparity inaccuracy because of low signal to noise ratio. On the other hand, the window size should be as small as possible to avoid the influence from the projection distortion. If the window size is too big and contains a significant variation of the disparity, it will cause the matching position incorrect due to the different projection distortions from the left and right images. In order to solve this problem, some researchers propose the weighting processing on the pixels within the fixed window, turning the

ISSN: 2005-4254 IJSIP

window selection problem into the weights assignment problem. In [17], a typical adoptive weight stereo matching algorithm is proposed. Other method is looking for the variable window for each pixel, where the method of Kanade and Okutomi[18] is a classic variable window stereo matching algorithm.

The method of self-adaptive variable window implicitly shows the selection criteria for the matching window size, which attracts the attention of many researchers. The reference [19] has proposed a kind of energy function to select the optimal window, considering the factors such as the variance, the average error, *etc.* within the window, and an effective optimization algorithm. The reference [20] uses the Gaussian distribution to describe the distribution of parallax and deduce the probability density function of parallax within the window, finally presenting the recurrence formula of parallax and the uncertainty function of window extended.

Compared with adaptive window algorithm, adaptive weighting algorithm is much easier to operate. The adaptive weighting framework proposed by Yoon *et al.* [17] has been widely applied. In this theoretical framework, the pixels in supporting window are supposed to make different matching contributions to center pixels, and the various pixels in supporting window should be assigned different weight values.

Yoon algorithm decreases the matching uncertainty while increases the matching precision. Wang *et al.* [21] improved Yoon algorithm through applying sigmoid function for optimizing the calculation of space weight value and introducing image segmentation method into improving the calculation of color weight value. In this paper, the advantages and disadvantages of sigmoid function in [21] will be analyzed firstly. Then neural network method is exploited to optimize the space weight value model. Image segmentation method is also used to calculate color weight value, and a new edge hold off Mean-Shift is proposed to improve the accuracy and reliability of segmentation. Finally, Loopy Belief Propagation algorithm is adopted to explore the optimal solution of measure function so as to get the disparity map.

2. Improved Cost Aggregation Function

2.1. Spatial Weight Algorithm Based on Neural Network

Stereo matching theory based on weight value believes that only if the supporting window enjoys the pixel with the same depth of central window, can it be used to support the similarity matching for central pixel. Because the depth information is unknown, without priori data, the only solution is to utilize the similarity of image color and spatial proximity to estimate the depth relation between different pixel points. Moreover, the characteristic of direct proportion between the weight value of central supporting match pixel and the probability of the weight value being in the same depth with central points is also used for calculation of weight value of supporting matching pixel.

Yoon weight function can be shown in formula (2).

$$w(p,q) = f_{\mathrm{c}}(\Delta c_{pq}) f_{\mathrm{c}}(\Delta g_{pq}) \tag{2}$$

Here, Δc_{pq} is the intensity space difference between pixel in supporting window and central pixel. Δg_{pq} is the spatial space distance.

In order to decrease the window size and calculation complexity, the only solution is to decrease a (a< 1) so as to get relatively small window cost, which exactly explains the reason why we adopt neural network in replace of Sigmoid space weight value. We use Sigmoid twice in NN space weight value model so as to ensure that the value of space weight value function will not be two small when a is relatively small, formulas (3)-(6) are the space weight value functions chosen in this section.

$$\begin{aligned} I_{\mathrm{M1}} &= x+b \\ I_{\mathrm{M2}} &= -x+b \\ I_{\mathrm{M3}} &= y+b \\ I_{\mathrm{M4}} &= -y+b \end{aligned} \tag{3}$$

$$O_{\mathrm{M}i} = f(I_{\mathrm{M}i}) \tag{4}$$

$$I_{\mathrm{T}} = \sum_{m=1}^{N} O_{\mathrm{M}m} - 3.5 \tag{5}$$

$$f_{\mathrm{c}}(\Delta g_{pq}) = f(I_{\mathrm{T}}) \tag{6}$$

Here, $f(x) = \dfrac{1}{1+e^{-ax}}$ is Sigmoid function.

2.2. Intensity Weight Function based on Edge Hold off Mean-Shift Algorithm

The intensity weight value function proposed by Yoon is shown as formula (2), however [21] proposes the below intensity weight value function based on segmentation theory. This theory holds that there exist same disparity in segmentation area, thus the intensity weight value should be 1.

$$f_{\mathrm{c}}\left(\Delta c_{pq}\right) = \begin{cases} 1 & p \in \mathrm{seg}(q) \\ \exp\left(-\dfrac{\Delta^2 c_{pq}}{\gamma_{\mathrm{c}}}\right) & \text{otherwise} \end{cases} \tag{7}$$

The segmentation algorithm based on PDE such as active contour, CV and others are not effective in image segmentation with complex background and time-consuming in calculation. What is more, the segmentation algorithm based on Markov needs determined segmentation number beforehand. However, Mean-Shift algorithm is estimation without variable which is easy and effective, and has become the most commonly used segmentation algorithm. Moreover, this algorithm can also function as image filter with high operation speed [22-23]. Whereas, Mean-Shift needs very rigorous selection of window, an improper selection of window will easily result in deficiency of edge and the result will be inaccurate. In this section, a new algorithm EHMean-Shift which combines edge hold off function will be demonstrated effective in edge protection.

For probability function $f_p(\boldsymbol{u}_i)$, there exist n d-dimensional space sampling points $\boldsymbol{u}_i$, the probability density of $f_p(\boldsymbol{u}_i)$ is estimated as

$$\hat{f}_p(\boldsymbol{u}) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{\boldsymbol{u}-\boldsymbol{u}_i}{h}\right\|^2\right) \tag{8}$$

Here, $k(x)$ is section function in which Mean-Shift vector can be expressed as

$$\boldsymbol{M}_{\mathrm{h}} = \frac{\sum_{i=1}^{n} (\boldsymbol{u}_i - \boldsymbol{u}) k\left(\left\| \frac{\boldsymbol{u} - \boldsymbol{u}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} k\left(\left\| \frac{\boldsymbol{u} - \boldsymbol{u}_i}{h} \right\|^2 \right)} \quad (9)$$

In formula (9), if $h = \alpha_{\mathrm{g}} h_0$, h_0 is the chosen fixed window size , $1 < \alpha_{\mathrm{g}} \le 1$ is a constant, $\alpha_{\mathrm{g}} = m(\|\nabla \boldsymbol{u}_0\|)$ is the function related to the gradient of $\boldsymbol{u}$ initial value. α_{g} is with the following characteristics: when *p* is peripheral point, gradient value is relatively big, thus $\alpha_{\mathrm{g}} \approx 0$. When *p* is not peripheral point, $\alpha_{\mathrm{g}} \approx 1$. From these, when *p* is peripheral point, $\alpha_{\mathrm{g}} h_0 \approx 0$, therefore it can be supposed that only point *p* itself is inside the window, thus $M_{\mathrm{h}} = 0$,the pixel of point *p* remains the same. It can be easily seen that the vector direction of EHMean-Shift is still the same with probability density gradient.

As for the function $m(x)$, we choose the below peripheral function

$$m(|\nabla \boldsymbol{u}_0|) = \frac{1}{1 + (|\nabla \boldsymbol{u}_0| / K)^2} \quad (10)$$

EHMean-Shift segmentation algorithm involves three phrases. The first phrase is Mean-Shift filtering phrase, in which some small segments are filtered for following segmentation. The second phrase is image segmentation. In the third phrase, small segments are removed. Pseudocode is shown as Fig. 2-4.

Filtering imge using EHMean-Shift
1. Read *img* Set h_{S} h_{I}
2. Computing gradient *g* for every pixel according to (14)~(18)
3. for *i*= 0 to *img.height* * *img.wight*
4. $(L_{\mathrm{n}}, u_{\mathrm{n}}, v_{\mathrm{n}}) = \mathrm{rgb2luv}(img(i))$ $x_{\mathrm{o}} = i.x$ $y_{\mathrm{o}} = i.y$
5. $\Delta = \mathrm{inf}$
6. Computing *m*(g(*i*)) using (12)
7. while($\Delta >$ threshold || *iter*<maxiter)
8. Computing kernel function $k_{\mathrm{all}} = k\left(\left\| \frac{\boldsymbol{u}^{\mathrm{S}}}{h_{\mathrm{S}}} \right\|^2 \right) k\left(\left\| \frac{\boldsymbol{u}^{\mathrm{I}}}{h_{\mathrm{I}}} \right\| \right)$ $M_{\mathrm{h}}^{\mathrm{S}}$ and $M_{\mathrm{h}}^{\mathrm{I}}$ for space and intensity according to (11)
9. $(x_{\mathrm{n}}, y_{\mathrm{n}}) = x_{\mathrm{o}} + m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{S}}$
$(L_{\mathrm{n}}, u_{\mathrm{n}}, v_{\mathrm{n}}) = (L_{\mathrm{o}}, u_{\mathrm{o}}, v_{\mathrm{o}}) + m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{I}}$
$(x_{\mathrm{o}}, y_{\mathrm{o}}, L_{\mathrm{o}}, u_{\mathrm{o}}, v_{\mathrm{o}}) = (x_{\mathrm{n}}, y_{\mathrm{n}}, L_{\mathrm{n}}, u_{\mathrm{n}}, v_{\mathrm{n}})$
10. $\Delta = \left\langle m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{S}}, m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{S}} \right\rangle + \left\langle m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{I}}, m(g(i))\boldsymbol{M}_{\mathrm{h}}^{\mathrm{I}} \right\rangle$
11. $img(i) = \mathrm{luv2rgb}(L_{\mathrm{n}}, u_{\mathrm{n}}, v_{\mathrm{n}})$

Figure 2. Pseudo-Code for 1st Stage


```
Separate image
1. Set label of every pixel as -1
   L = -1
2. for i= 0 to img.height * img.wight
3.   if label(i) < 0
4.   push i to stack
     Label(i) = ++L;
5.   while(stack is not empty)
6.     currentpixel= topstack;
7.     pop currentpixel from stack
8.     search 8 directions of currentpixel
9.     if
|img(neighbor(currentpixel)) − img(i)| < colorthreshold
10.        label(neighbor(currentpixel)) = L
11.        push neighbor(currentpixel) to stack
12.    Set intensity(L) as average intensity of pixels which belong to label L
13. Establish linked lists for all clusters
```

Figure 3. Pseudo-Code for 2nd Stage

```
Prune
1. for i = 0 to maxlabel
2.   if number(i) <minLabel
3.      delete label(i)
4. Reestablish linked lists for all clusters
```

Figure 4. Pseudo-Code for 3rd Stage

2.3. Aggregation Cost

In order to enhance the matching accurate, the Census transformation is introduced to build the cost function, the final improved cost aggregation function is expressed as below:

$$\bar{E}(p,d) = \frac{\sum_{q\in W(p)} \left(M(q,d) + dsim_{\mathrm{CT}}(q,d)\right)}{\left\|W(p)\right\|} \tag{11}$$

Here, $\left\|W(p)\right\|$ is the number of supporting points inside window, $M(q,d)$ is computed by traditional adaptive weight method.

$$M(p,d) = \frac{\sum_{q\in N_p} w(p,q)w(p-d,q-d)dsim(q,q-d)}{\sum_{q\in N_p} w(p,q)w(p-d,q-d)} \tag{12}$$

$w(p,q)$ shows the weight value assigned to pixel p, while q represents the pixel in supporting window of central pixel p. $w(p-d,q-d)$ possesses the same definition with $w(p,q)$, representing the weight value of pixel in supporting window of point $p-d$ corresponded with point p in right image. $dsim(q,d)$ in $M(q,d)$ adopts the similarity measure function proposed by Birchfiled.

Hamming Distance is applied for matching cost calculation. Comparison of corresponding pixel Census transformation can determine the different numbers of strings.

A small Hamming Distance means a high similarity of corresponding pixel. Census transformation cost can be expressed as formula (13).

$$dsim_{CT}\left(p,d\right)=\left\|CT\left(p\right)\oplus CT\left(p-d\right)\right\|_1 \tag{13}$$

Here, $\left\|\bullet\right\|_1$ represents the number of 1 in calculated strings. $CT\left(p\right)$ is the census transform.

3. LBP Algorithm

Loopy Belief Propagation is a global optimization algorithm proposed by J. Pearl [24], which can be optimized in stereo matching algorithm as an energy function as formula (14).

$$E(f)=\sum_{p\in P}D_p(f_p)+\sum_{(p,q)\in N}V(f_p-f_q) \tag{14}$$

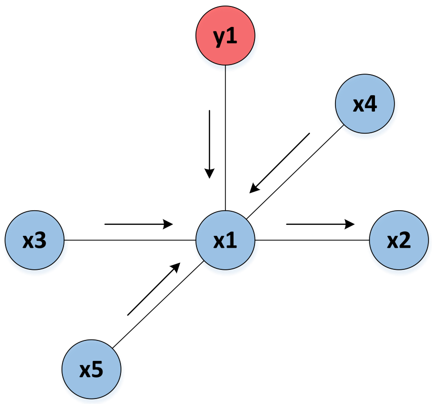

Figure 5. Message Passing in LBP Algorithm.

LBP is an algorithm based on message sending in which nodes in the network change information with neighboring nodes to calculate its probability condition. Fig. 5 is the LBP algorithm message sending scheme, x_1 □ x_5 are hidden variants, namely parallax of this point, while y_1 is observed variant, namely the observed value of chromaticity. When x_1 receives message from neighboring nodes except for x_2, x_1 will send message to x_2. Take Max-Pruduct for instance the message format is shown as formula (15)

$$\begin{aligned} msg_{x_1\to x_2}\left(l_{x_2}\right)=\min_{l_{x_1}}\Big(D_{x_1}\left(l_{x_1}\right)+V\left(l_{x_1},l_{x_2}\right)\Big) \\ +\sum_{k\in N(x_1)\backslash x_2} msg_{k\to x_1}\left(l_{x1}\right) \end{aligned} \tag{15}$$

After several circulation, the below formula is adopted to calculate the label of the minimum value of every node, namely parallax value.

$$\arg\min_{l_{x_1}}\left(D_{x_1}\left(l_{x_1}\right)+\sum_{k\in N(x_1)} msg_{k\to x_1}\left(l_{x_1}\right)\right) \tag{16}$$

In order to improve the accuracy of LBP algorithm, image pyramid [25] is applied in reference [26] to LBP algorithm. The original images are samples on five levels and the parallax image is obtained on sub- pixel level, which benefits a lot tfor getting accurate parallax image.

4. Simulation Experiment and Discussions

This section firstly check the influence of variants on performance, then benchmarks provided through Middlebury database are used to evaluate the performance of this algorithm.

4.1. Effect of Parameter a on Performances of Our Algorithm

Main text paragraph This section will take Tsukuba and Venus for example to discuss on the influence of *a* on the algorithm. Major variants chosen in this section include $h_S = 5$， $h_I = 6.5$， $\gamma_c = 5$， *b*=3， *w*=23， *K*=5， Error Threshold=1. The simulation result is shown in Fig. 6.

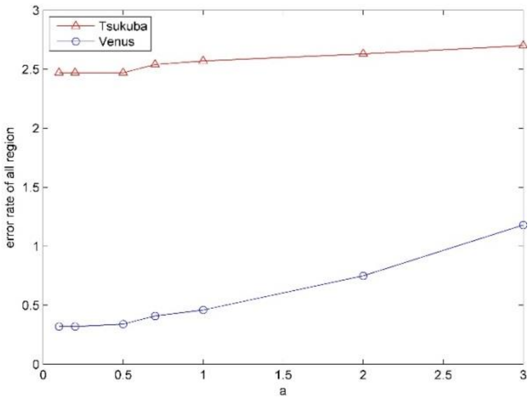

Figure 6. Performance Evaluation (Disparity Error Rates for All Regions) of The Proposed Method when Varying the Parameter Values

From the above figure: (1) the decrease of a can improve matching accuracy and reduce matching error rate, which is identical to the analysis of the influence of *a* on measure function; (2) when *a*<0.5, their performance is nearly identical, the comprehensive performance and algorithm operating time window can be valued to 23. When *a* =0.1, the result with low matching error rate will be produced.

4.2. Influence of Segmentation Algorithm in Performance

This section will discuss on the influence of segmentation algorithm on matching algorithm performance. For example, when window is 23, other variants remain unchanged, the matching error rate of nonocc, all and disc are shown in below Fig.s 7-9. In order to exclude the influence of other factors, post-processing programs are not operated in this simulation, and only the performances of initial parallax images are compared. In segmentation algorithm, $h_S = 10$ ， $h_I = 7$. From below figures, the disparity map obtained from EHMean-Shift segmentation algorithm, is higher in accuracy than Mean-Shift, the matching error rates of nonocc, all and disc reduces by 2.82%, 1.29% and 3.24% respectively.

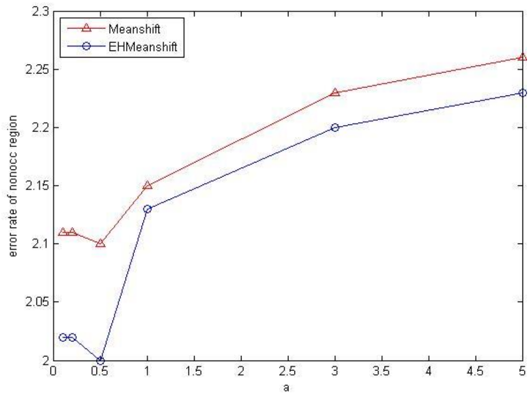

Figure 7. Comparison Between Error Matching Rate of Nonocc Region Obtained by Our Algorithm with EHMean-Shift and Mean-Shift.

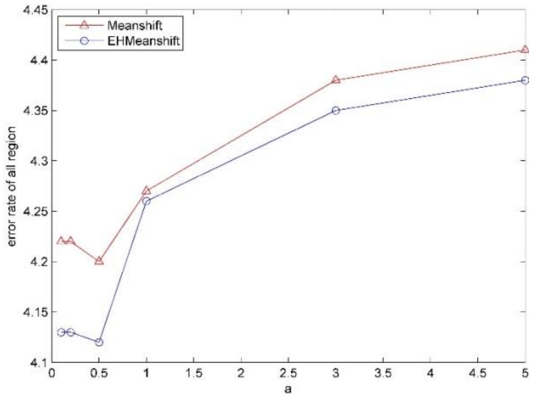

Figure 8. Comparison Between Error Matching Rate of All Region Obtained by Our Algorithm with EHMean-Shift and Mean-Shift.

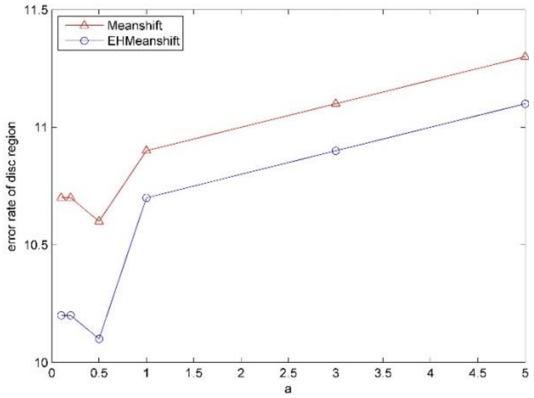

Figure 9. Comparison Between Error Matching Rate of Disc Region Obtained by Our Algorithm with EHMean-Shift and Mean-Shift.

4.3. Comparison with Yoon Algorithm

Stereo matching algorithm proposed by Yoon is a classical adaptive weight stereo matching algorithm. One of the research objectives of this section is to reduce the size of matching window and calculation load. The values in this section are set as follows:

$h_S = 5$, $h_I = 6.5$, window size is 23, window size of Yoon algorithm is 35. The comparison result can be found in table 1.

Table 1. Comparison of Performance Between our Method and Yoon's Method

Algorithm	Avg. Rank	Tsukuba			Venus		
		nonocc	all	disc	nonocc	all	disc
AdaptWeight	78.5	1.38_{47}	1.85_{50}	6.90_{55}	0.71_{93}	1.19_{97}	6.13_{97}
Our Method	31.5	1.11_{20}	2.42_{75}	5.74_{21}	0.13_{9}	0.32_{16}	1.75_{16}

Teddy			Cones			Average percent of bad pixels
nonocc	all	disc	nonocc	all	disc	
7.88_{92}	13.3_{88}	18.6_{100}	3.97_{87}	9.79_{82}	8.26_{54}	6.67
5.58_{33}	10.8_{38}	14.8_{40}	2.80_{40}	8.35_{38}	7.60_{32}	5.12

From this table, the average error pixel percentage is 5.12 in the algorithm proposed in this paper while that in Yoon algorithm is 6.67 with 23.24% reduced. The matching error rate of nonocc, all and disc all show to be lower than Yoon algorithm. Moreover, the optimal window size calculated in this paper is 23, however this value is 35 in Yoon algorithm. It can be concluded that the algorithm proposed in this chapter realizes the design objective by using small window to get good matching effect.

4.4. LBP Algorithm Energy Change

This section is to compare the change of energy in standard LBP, LBP proposed in [33] and formula (14) proposed in this paper. The measure function in simulation is shown as formula (17)

$$dsim = \lambda \min(\left|I_L(p) - I_L(p-d)\right|, TH_I) \tag{17}$$

Here, $\lambda = 0.07$, $TH_I = 15$, The relevant variants in the algorithm proposed in this paper include: window size 5 and 11, *a*=0.1, others unchanged, the simulation result is shown in Fig. 10. From the figure, the initial energy in standard LBP algorithm is 1.0083×10^7, after eight circulations, the energy is reduced to 1.064×10^6, still much bigger than in other three methods. Thus, standard LBP algorithm needs a lot of circulation to get ideal results. The algorithm proposed in this paper can get rather accurate disparity map with less circulation (only three) than [33]. In this paper, the pixel similarity function has been refined in this paper so as to reduce energy consumption and get accurate disparity map by less circulation, significantly improving the operating efficiency.

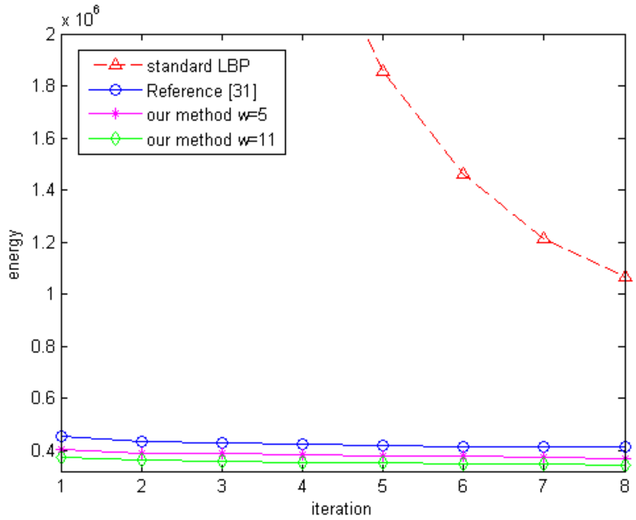

Figure 10. Diagram of Energy Change

4.5. Performance Evaluation

This section uses Tsukuba, Venus, Teddy and Cones in Middlebury database to evaluate on the performance of algorithm. Similar to above analysis, our performance evaluation focuses on the matching error rate of nonocc, all and disc. The simulation window is 31, other variants remain unchanged. Table 2 shows the matching error rate of the algorithm proposed in this paper in Middlebury database. This table also lists the matching error rates of this algorithm in different areas.

Table 2. The Error Percentages in Different Regions Area

Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc
Surface Stereo[27]	28.6	1.28_{39}	1.65_{28}	6.78_{49}	0.19_{27}	0.28_{13}	2.61_{49}
LLR[28]	30.2	1.05_{14}	1.65_{27}	5.64_{16}	0.29_{61}	0.81_{76}	3.07_{61}
Our Method	31.5	1.11_{20}	2.42_{75}	5.74_{21}	0.13_{9}	0.32_{16}	1.75_{16}
MultiAgg[29]	32.2	1.52_{61}	1.82_{42}	8.20_{81}	0.16_{19}	0.39_{32}	2.03_{26}
WarpMat[30]	32.2	1.16_{23}	1.35_{8}	6.04_{31}	0.18_{20}	0.24_{9}	2.44_{40}

Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc	Average percent of bad pixels
3.12_{6}	5.10_{2}	8.65_{3}	2.89_{44}	7.95_{28}	8.26_{55}	4.06
4.56_{17}	9.81_{29}	12.2_{17}	2.17_{7}	8.02_{29}	6.42_{8}	4.64
5.58_{33}	10.8_{38}	14.8_{40}	2.80_{40}	8.35_{38}	7.60_{32}	5.12
5.09_{24}	10.5_{34}	13.8_{28}	2.27_{11}	7.49_{15}	6.71_{13}	5.00
5.02_{22}	9.30_{27}	13.0_{24}	3.49_{65}	8.47_{43}	9.01_{69}	4.98

5. Conclusions

Traditional adaptive weight value needs rather big window and large calculation load to generate accurate disparity map. In order to decrease the calculation load, this paper improves the space weight value and chromaticity weight value in Yoon algorithm, so as to ensure their good performance in small window. The entire matching algorithm utilizes the improved method and conduct matching cost aggregation, and adopts LBP based on image pyramid to explore global optimization.

(1) Decreasing value of a can improve matching accuracy and lower matching error rate. When a<0.5, there does not exist significant difference between performance of different algorithms, we choose 0.1 here;

(2) Compared with traditional Mean-Shift, EHMean-Shift proposed in this paper can help the matching error rates of nonocc, all and disc reduce by 2.82%, 1.29% and 3.24% respectively;

(3) Compared with Yoon algorithm, the matching error rates of nonocc, all and disc of four picture in Middlebury database shows that the algorithm proposed in this paper with window size 23 is with higher performance than Yoon algorithm proposed with window size 35, the average matching error rate reducing 23.24%;

(4) The algorithm proposed in this paper significantly reduces the energy consumption of LBP. Thus the circulation number of LBP in this algorithm can be smaller in value.

Acknowledgements

This work was supported by Shandong Natural Science Foundation under Grant ZR2013FL033. This work was also supported in part by Shandong Provincial Education Department under the international cooperation program for key professors of 2014, and Zaozhuang University Doctor Science Foundation.

References

[1] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in Proc. IEEE CVPR, Minneapolis, MN, USA, **(2007),** pp. 1–8.

[2] J. C. Kim, K. M. Lee, B. T. Choi and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," in Proc. IEEE CVPR, San Diego, CA, USA, **(2005)**, pp. 1075–1082.

[3] C. Van Meerbergen, M. Vergauwen, M. Pollefeys and L. Van Gool, "A hierarchical symmetric stereo algorithm using dynamic programming," International Journal of Computer Vision, vol. 47, no. 1–3, , **(2002)**, pp. 275–285.

[4] C. S. Park and H. W. Park, "A robust stereo disparity estimation using adaptive window search and dynamic programming search," Pattern Recognition, vol. 34, **(2001)**, pp. 2573–2576.

[5] C. Leung, B. Appleton and C. Sun, "Iterated dynamic programming and quadtree subregioning for fast stereo matching," Image and Vision Computing, vol. 26, no. 10, **(2008)**, pp. 1371–1383.

[6] V. Kolmogorov and R. Zabih, "What energy can be minimized via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 2, **(2004)**, pp. 147–159.

[7] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, **(2001)**, pp. 1222–1239.

[8] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts," in Proc. IEEE CVPR, Washington, DC, USA, **(2004)**, pp. 437–444.

[9] Y. Bovkov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, **(2004)**, pp. 1124–1137.

[10] J. Sun, Y. Li, S.-B. Kang and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in Proc. IEEE CVPR, San Diego, CA, USA, **(2005)**, pp. 399–406.

[11] J. Sun, N.-N. Zheng and H.-Y. Shum, "Stereo matching using belief propagation," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, **(2003)**, pp. 787–800.

[12] Q.-X. Yang, L. Wang, R.-G. Yang, H. Stewenius and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 3, **(2009)**, pp. 492–504.

[13] A. T. Ihler, J. W. F. Iii and A. S. Willsky, "Loopy belief propagation: convergence and effects of message errors," Journal of Machine Learning Research, vol. 6, **(2005)**, pp. 905–936.

[14] A. Bruhn, J. Weickert, C. Feddern and T. Kohlberger, "Variational optical flow computation in real time," IEEE Transactions on Image Processing, vol. 14, no. 5, **(2005)**, pp. 608–615.

[15] R. Ben-Ari and N. Sochen, “Variational stereo vision with sharp discontinuities and occlusion handling,” in Proc. IEEE ICCV, Rio de Janeiro, Brazil, (**2007**), pp. 1–7.
[16] R. Ben-Ari and N. Sochen, “Stereo matching with Mumford-Shah regularization and occlusion handling,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 11, (**2010**), pp. 2071–2084.
[17] K. J. Yoon and I. S. Kweon, “Adaptive support-weight approach for correspondence search,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, **(2006)**, pp. 650–656.
[18] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: theory and experiment,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 9, (**1994)**, pp. 920–932.
[19] O. Veksler, “Fast variable window for stereo correspondence using integral images,” in Proc. IEEE CVPR, Madison, WI, USA, (**2003)**, pp. 556–561.
[20] M. Okutomi, T. Kanade, “A locally adaptive window for signal matching,” in Proc. IEEE ICCV, Osaka, Japan, (**1990)**, pp. 190–199.
[21] W. Fuzhi and H. Dagu, “Improved Yoon stereo matching algorithm based on adaptive weight,” Journal of Electronic Measurement and Instrument, vol. 24, no. 7, (**2010)**, pp. 632-637.
[22] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, (**2002)**, pp. 603-619.
[23] D. Comaniciu, V. Ramesh and P. Meer, “The variable bandwidth mean shift and data-driven scale selection,” in Proc. IEEE ICCV, Vancouver, BC, **(2001)**, pp.438-445.
[24] A. T. Ihler and A. S. Willsky, “Loopy belief propagation: convergence and effects of message errors,” Journal of Machine Learning Research, no. 6, (**2005)**, pp. 905- 936.
[25] G. Bradski and A. Kaebler, Learning OpenCV. Sebastopol, CA, USA: O'Reilly Media, (**2008)**.
[26] P. F. Felzenszwalb and D.P. Huttenlocher, “Efficient belief propagation for early vision,” International Journal of Computer Vision, vol. 70, no. 1 (**2006)**, pp. 41–54.
[27] M. Bleyer, C. Rother and P. Kohli, “Surface stereo with soft segmentation,” in Proc. IEEE CVPR, San Francisco, CA, (**2010)**, pp.1570-1577.
[28] S. Zhu, L. Zhang and H. Jin, “A locally linear regression model for boundary preserving regularization in stereo matching,” in Proc. ECCV, Florence, Italy, **(2012)**, pp.101-115.
[29] M. Bleyer, M. Gelautz, C. Rother and C. Rhemann, “A stereo approach that handles the matting problem via image warping,” in Proc. IEEE CVPR, Miami, FL, (**2009)**, pp. 501-508.
[30] X. Tan, C. Sun, D. Wang, Y. Guo and T. D. Pham, “Soft cost aggregation with multi-resolution fusion,” in Proc. IEEE CVPR, Zurich, Switzerland, (**2014)**, pp. 17-32.

Authors

Zheng Sun received the B.S. in electronic information engineering from Qingdao University, China, in 2005, and the Ph.D. in mechatronic engineering from China University of Mining and Technology, China, in 2010. From 2010 to 2013, he was a Lecturer with the School of Mechatronic Engineering, Zaozhuang University. Since 2014, he has been an Associate Professor with the School of Mechatronic Engineering, Zaozhuang University. From Dec. 2014 to May 2015, he was a Visiting Associate Professor with the Department of Electrical and Computer Engineering, McMaster University.He is the author of more than 20 articles. His research interests include vision inspection, vision navigation, wireless sensor networks, and industrial automation.

Rong Yang received her B.S. and M.S. degrees from Department of mechanical engineering at Xi’an Jiaotong University, China, in 2010 and 2012, respectively. She is currently a Ph.D. candidate in department of mechanical engineering at McMaster University, Canada. Her main research areas include electric vehicle design, interior permanent magnet motor design and electric machine manufacturing.

Hui Li received her B.S. and M.S. degrees from College of mechatronic engineering at China University of Mining and Technology, China, in 2005 and 2008, respectively. He is currently a Ph.D. candidate in College of mechatronic engineering at China University of Mining and Technology, China.His research interests include vision navigation in robot vision, and wireless sensor networks.

