ADAPTIVE STEREO MATCHING ALGORITHM BASED ON EDGE DETECTION

Kun Wang

The Institute of Artificial Intelligence and Robotics Xi'an Jiao Tong University China
kwang@aiar.xjtu.edu.cn

ABSTRACT

It is a difficult problem to choose appropriate image windows in area-based stereo matching algorithm. Most adaptive window algorithms utilize an evaluation function with disparity and intensity to choose the appropriate windows. This paper presents a new adaptive stereo matching algorithm based on edge detection. It chooses the appropriate image window only by the intensity variance and the process has no relation with disparities. This is the main difference between this algorithm and the others. Meanwhile we introduce a new rank transform method into the stereo algorithm. To demonstrate the effectiveness of the algorithm, it has been tested by stereo images with ground truth in Middlebury stereo database. The results show that our algorithm has high correct matching rate. Meanwhile it is robust and has few parameters to be set.

1. INTRODUCTION

The most important problem for the area-based stereo algorithm is the selection of image windows. The fixed window often doesn't work well for the whole image. In order to get reliable disparity information, different window size should be taken at different pixels in the same image. According to the results of [1][4], reliable windows should be large enough to include enough intensity variation for reliable matching but small enough to cover only pixels at equal depth. In general, large window may include more intensity variation information for disambiguation and it will get reliable results especially in low texture areas. But at the depth discontinuity boundary, large windows may overlap discontinuity areas and blur the boundaries.

2. ADAPTIVE WINDOW VIA EDGE DETECTION

In recently years, several adaptive stereo matching algorithms were proposed such as [1][4][5][6]. Kanade and Okutomi [4] employed a statistical model of disparity distribution within a window. Olga Veksler proposed the compact window algorithm [1]. These adaptive algorithms constructed an evaluation function of disparity and intensity to choose the appropriate window for each pixel. It is difficult to construct the evaluation function with the disparity and intensity. And the efficient optimization over all kinds of windows is also a problem.

This paper presents a new adaptive algorithm that chooses the appropriate window only by intensity variation and this process has no relation with the disparity. As we all know, large windows may get reliable results in low texture areas and blur the depth discontinuity boundary. But if the large window only covers the pixels belonging to the same depth, the boundary will not be blurred. It means that we should detect the depth discontinuity boundary. But it is very difficult to detect the depth boundary directly.

In real images, the depth boundary often causes acute intensity variance. They are contained within the intensity edges. Our algorithm finds out the pixels that have great intensity variance with edge detection method. In low texture areas, there are few edge pixels, so we should take large windows to include enough intensity variance. At depth discontinuity boundaries there are many edge pixels. Small windows should be taken to avoid covering areas with different depth. When a pixel belongs to edges but it does not belong to the depth boundaries, it must be in the high textured areas. A small window is also appropriate for it. Moreover if we could enlarge the window to the direction that has the same depth, it will not only avoid overlapping the depth boundary but also include more intensity variance information for disambiguation. So it is feasible to make the adaptive window selection only by the intensity variance. Details are introduced as follows.

Firstly, we compute the intensity edges with edge detection algorithm and save the results in a two dimensional matrix *edge(x, y)*. The depth boundaries are obvious edges in real images, and most of the edge detection algorithms could find them out. Our algorithm only needs these obvious clews. We have used different edge detection algorithm from simple threshold algorithm to Canny detector and the correct matching rate does not changed much.

Fig.1 shows an example of the left image of Tsukuba. The depth boundaries are contained within the white pixels. Although some white pixels do not belong to the depth boundary, small image windows are also

0-7803-8554-3/04/$20.00 ©2004 IEEE.

appropriate for these high textured areas. The matching results will not be affected. Based on the matrix *edge(x, y)*, the algorithm could choose the appropriate image window for every pixel as follows.

1) For every pixel *(x, y)*, the window size is started from 3*3 (*r=3*). We propose an evaluation function (1) to choose the image window.

$$E_r(x,y)=\sum_{\substack{p\in[x-(r-1)/2,\,x+(r-1)/2],\\ q\in[y-(r-1)/2,\,y+(r-1)/2]}} edge(p,q)$$

where r is the window size (1)

2) If $E_r(x,y)$ is greater than m, the window size 3*3 is appropriate for the pixel *(x, y)*. Then turns to the step 4).

3) If $E_r(x,y)$ is less than m, then enlarge the image window (*r=r+2*). And compute the $E_r(x,y)$ again with the new image window and compare it with n. If $E_r(x,y)$ is less than n, enlarge the window size until it is greater than n.

4) Add a column at the left border of the image window, and compute $E_r(x,y)$ with the new window. If it is not increased, then add another column at the left border until it is increased. If $E_r(x,y)$ is increased, cancel the last added column.

5) Do similar operations at the other three sides of the image window and save the results.

The value of $E_r(x,y)$ represents the intensity variance in the image window. When the window size is 3*3 and $E_r(x,y)$ is greater than 3, it means that the window is exactly on the variance edges and it should not be enlarged. So we let m=3. The threshold n works when the image window is enlarged in low texture areas. When $E_r(x,y)>n$, we consider the image window is enlarged to the edges or has enough texture. So we let n=1. The step 4) and 5) will enlarge the image window to the direction that has the same depth with the observed pixel. It is very useful to the pixels at the discontinuity boundary for example *w7* and *w8* in Fig. 2. It enables to get enough intensity variance while only covering the pixels that belong to the same depth.

Our adaptive window method is presented heuristically. There is no more theoretical discussion to motivate the rules that are proposed to select the window size. But the effectiveness of this method can be proved by the correct matching rate of the depth map.

Fig. 2 shows some windows taken by our algorithm. Large windows are taken in low texture areas such as *w3* and *w5*. And small windows are taken in the depth boundaries for example *w2* and *w6*. Moreover the windows near the depth boundaries taken by our algorithm are continuous. It is a good property as mentioned in the future work part of paper [8]. In this paper the image windows are all rectangle windows but the pixel in research may not be at the center of the rectangle window, for example, *w3* and *w8*. The black pixel in the white window is the observed pixel.

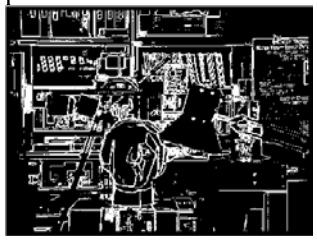

Fig.1 the edges

Fig.2 some examples

3. IMPLEMENTATION OF THE ALGORITHM

There are four main steps in the algorithm. The first step is adaptive window selection that is introduced in section 2. The second step is rank transform. The third step is feature calculation and the fourth step is matching calculation and disparity selection.

3.1. rank transform

This paper extends the rank transform [3] ideas and proposes a new rank transform equation to solve the problems such as image noise and brightness difference in stereo images.

Calculate the intensity difference between the center pixel *(x, y)* and other pixels in the feature window. According to the intensity difference we define five memberships for every pixel. They are approximate level, bigger level, biggest level, smaller level and smallest level. It is shown in equation (2) and *dif(x+i, y+j)* is the intensity difference between pixel *(x+i, y+j)* and pixel *(x, y)*. *fuz(i, j)* denote the value after rank transform. With some experiments, we find that when the number of ranks is greater equal than 5, the correct matching rate does not have obvious increase. So we choose 5 memberships. While in [3] it counts the number of pixels in the image window whose values is less than the center pixel's. [3] is a particular case of our method.

$$fuz(i,j)=\begin{cases}-2 & \text{if } dif(x+i,y+j)<-s \quad \text{smallest level}\\ -1 & \text{if } -s\le dif(x+i,y+j)<-t \quad \text{smaller level}\\ 0 & \text{if } -t\le dif(x+i,y+j)\le t \quad \text{approximate level}\\ 1 & \text{if } t<dif(x+i,y+j)\le s \quad \text{bigger level}\\ 2 & \text{if } s<dif(x+i,y+j) \quad \text{biggest level}\end{cases} \tag{2}$$

The parameters s and t are determined by the rules that they should decrease the effect of the large noise and permit the disturbance of the little noise. In experiments we let the threshold t=2 and s=9.

Why do we use the rank transform method? Firstly, it could solve the brightness difference between the stereo image pairs. Fig. 3 shows a simple example that the left and right image windows consist only 3 pixels. Fig. 3a [1] is the perfect status and it is easy to find the matching pixels. In Fig. 3b the right image is brighter than the left image. If we use the SAD or SSD methods in Fig. 3b [1], we will find even in matching pixel there is great difference. It is difficult to make the correct discrimination locally. The results of rank transform of Fig 3b are shown in Fig. 3c. And the matching task in Fig. 3c is as easy to be accomplished as in Fig 3a. Secondly, most of the area-based methods are sensitive to noise because they are based on the local information. Rank transform method could solve the problem. The large noise often has great intensity difference compared with other pixels. In SAD or SSD algorithm, a great noise would disturb the matching results. But after the rank transform such as the equation (2), most of the large noises are transformed into the greatest or smallest level. So it only disturbs one pixel and has little effect on the matching results.

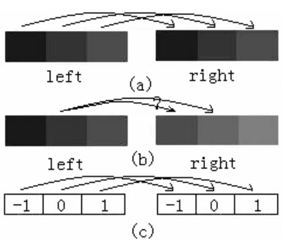

Fig. 3 (a)perfect status (b)light difference (c) rank transform results

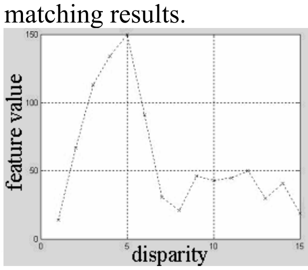

Fig. 4 the feature value of pixel (87,338) in "Tsukuba"

3.2. feature matrix

We use $f\_mat_d(x, y)$ to denote the similarity between the pixel $(x\text{-}d, y)$ in the right image and the pixel (x, y) in the left image. Suppose the observed pixel are (x, y) in left image and its matching pixel $(x\text{-}d, y)$ in right image. We take rank transform for the two pixels and get two matrixes $fuz_l(p, q)$ and $fuz_r(p, q)$. Then calculate the number of the elements in $fuz_l(p, q)$ which has the same rank membership with the elements in $fuz_r(p, q)$ and save it in $f\_mat_d(x, y)$ as features. Do this step for every pixel and get the matrix f_mat_d. It is the feature matrix when the disparity value is d. The feature value of pixel (87, 338) in image "Tsukuba" with different disparities is shown in Fig. 4. The x axis is the disparity value and the y axis is the feature value. It has a dominant maximum at the disparity 5 that corresponds to the correct matching disparity.

3.3. match calculation

Calculate the feature matrix for every disparity from 0 to $d_{\max}$, and computes the match value $match_d(x, y)$ with equation (3). The size of match window is $mwin\_x{*}\, mwin\_y$. Then find the maximum match value among different disparity with equation (4) and the disparity of the maximum $match_d(x, y)$ is the disparity of pixel (x, y).

$$match_d(x, y) = \sum_{\substack{p\in[xlow,xup]\\ q\in[ylow,yup]}} f\_mat_d(p, q)$$

where
$$xlow = x - mwin\_x/2, \quad xup = x + mwin\_x/2$$
$$ylow = y - mwin\_y/2, \quad yup = y + mwin\_y/2 \qquad (3)$$

$$disparity(x, y) = \arg\max_d \{match_d(x, y)\} \qquad (4)$$

4. EXPERIMENTAL RESULTS

Some grayscale stereo pairs are used to demonstrate the effectiveness of the presented algorithm. Four of them are downloaded from www.middlebury.edu/stereo. Our algorithm works well with the same parameters for all image pairs we tried.

Fig. 5a is the ground truth of "Tsukuba". There are homogeneous and low texture areas in the image, for example the upper right area of the image. Fig. 5b is the result of our algorithm. There are 97.14% of disparities that are off the ground truth within ± 1. The error rate is 2.86%, while the error rate of the compact window algorithm [1] is 3.36%. We have evaluated the performance of our algorithm at www.middlebury.edu/stereo [2]. Table 1 shows the performance table of the top 11 algorithms. The results in Table 1 show that our algorithm outperforms the compact window algorithm [1]. And compared with the new variable window algorithm [8] that is the best local algorithm now, our algorithm is comparable to it in correct matching rate. But it is slowly than [8]. While compared with global methods such as [7] and [9], our algorithm gives less exact disparity map. But it has no prior on disparity map and it is faster than most of the global algorithm. Fig.7 is the Venus images in which our algorithm owns the highest correct matching rate at Table 1.

The results show that our algorithm produces accurate disparity map and could properly handle noise and low texture areas. It is robust with respect to parameters that it does not need to be tuned. While in Fig.5c, the algorithm does not properly deal with the homogeneous areas in the upper left of Tsukuba. The problem will be solved in the future work.

5. CONCLUSONS

This paper proposes a new rank adaptive stereo matching algorithm based on edge detection. The main feature of the algorithm is that it selects the adaptive window only by intensity variance. Meanwhile the algorithm successfully puts a new rank transform algorithm into stereo matching. It doesn't use any prior information about images. The results show that this algorithm can recover the depth boundaries precisely.

The author would like to give thanks to Dr. Daniel Scharstein and Richard Szeliski for the stereo images with ground truth and evaluation. This work is supported by NSF of China under grants of 60205001 and 60021302.

6. REFERENCES

[1] Olga Veksler, "Stereo Matching by compact windows via minimum cycle", ICCV'01, pp.540-547, 2001

[2] D. Scharstein and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", IJCV 47(1/2/3):7-42, April-June 2002.

[3] Jasmine Banks, and Mohammed Bennamoun, "Reliability Analysis of the Rank Transform for Stereo Matching ", IEEE Trans. Syst., Man,Cyber., vol. 31.pp.870-880,Dec 2001

[4] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and Experiment," IEEE Trans. PAMI, vol. 16. No. 9, pp.920-932, Sept. 1994

[5] Masatoshi Okutomi and Yasuhiro Katayama, "A Simple Algorithm to Recover Precise Object Boundaries and Smooth Surfaces", Proceedings of the SMBV'01, 2001

[6] Maas, R., Haar Romeny, B.M.; Viergever, M.A, "Area-based computation of stereo disparity with model-based window size selection", CVPR 1999, vol.: 1

[7] Yuri Boykov, Olga Veksler, Ramin Zabih, "Fast Approximate Energy Minimization via Graph Cuts", IEEE Trans. PAMI, vol. 23, no. 11, pp.1222-1239, November 2001.

[8] O. Veksler, "Fast Variable Window for Stereo Correspondence using Integral Images" CVPR 2003.

[9] J. Sun, H. Y. Shum, and N. N. Zheng, "Stereo matching using belief propagation" ECCV 2002.

Algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untex.	disc.	all	untex.	disc.	all	untex.	disc.	all	disc.
Layered	1.58 4	1.06 6	8.82 5	0.34 1	0.00 1	3.35 1	1.52 8	2.96 17	2.62 2	0.37 10	5.24 10
MultiCam GC	1.85 7	1.94 12	6.99 4	0.62 5	0.00 1	6.86 9	1.21 6	1.96 8	5.71 6	0.31 7	4.34 9
Belief prop.	1.15 1	0.42 2	6.31 1	0.98 7	0.301 2	4.83 5	1.00 4	0.76 4	9.13 11	0.841 7	5.27 11
GC+occl.	1.19 2	0.23 1	6.71 2	0.73 6	0.11 7	5.71 7	1.64 11	2.75 15	5.41 5	0.611 3	6.05 12
Impr. Coop.	1.67 5	0.77 4	9.67 8	1.21 11	0.17 10	6.90 10	1.04 5	1.07 5	13.68 16	0.29 5	3.65 6
Disc. pres.	1.78 6	1.22 8	9.71 9	1.17 9	0.08 6	5.55 6	1.61 10	2.25 11	9.06 10	0.32 8	3.33 5
GC+occl.	1.27 3	0.43 3	6.90 3	0.36 2	0.00 1	3.65 2	2.79 19	5.39 20	2.54 1	1.79 20	10.08 2
Graph cuts	1.94 9	1.09 7	9.49 7	1.30 13	0.06 5	6.34 8	1.79 14	2.61 14	6.91 7	0.31 6	3.88 7
Symbiotic	2.87 12	1.71 10	11.90 10	1.04 8	0.13 8	7.32 12	0.51 2	0.23 2	7.88 8	0.50 12	6.54 13
Var. win.	2.35 10	1.65 9	12.17 12	1.28 12	0.23 11	7.09 11	1.23 7	1.16 6	13.35 15	0.24 3	2.98 3
Our method	**2.86 11**	**1.87 11**	**13.76 16**	**1.21 10**	**0.02 4**	**7.72 13**	**0.44 1**	**0.14 1**	**4.30 3**	**0.68 15**	**8.89 16**

Table 1 the performance of different stereo algorithms

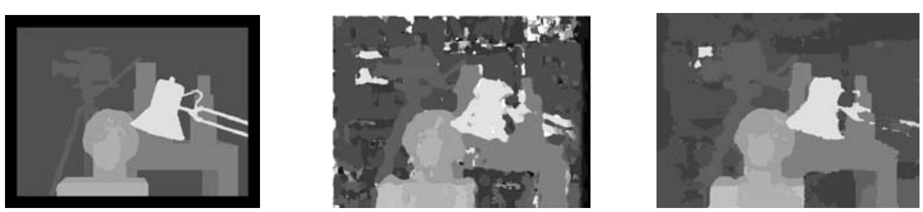

Fig.5 (a) ground truth of "Tsukuba" (b) the result with fixed windows (c) result of the algorithm

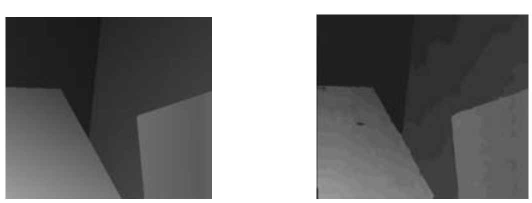

Fig. 6 (a) ground truth of Venus (b) result of our algorithm