Disparity Estimation Using Edge Preserving Method

Yong-Jun Chang and Yo-Sung Ho
School of Information and Communications
Gwangju Institute of Science and Technology (GIST)
123 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Republic of Korea
{yjchang, hoyo}@gist.ac.kr

***Abstract*—A disparity map is usually obtained by stereo images using a stereo matching method. Edge boundaries in the disparity map separate two different objects. Therefore, edge preservation is one of important issues in the stereo matching method. A conventional distance transform was proposed to preserve edge boundaries in the disparity map. However, this method has a complexity problem because of its iterative calculation. To reduce the calculation complexity, we propose a modified distance transform. Another important issue in the stereo matching method is an occlusion handling. We improve the quality of occlusion regions using disparity values of neighboring pixels. We apply the occlusion handling method to the edge preserved disparity map. Experiment results show that the proposed method is faster than the conventional method with a small loss in bad pixel rates.**

Keywords—stereo matching; distance transform; occlusion handling

I. INTRODUCTION

Three-dimensional (3D) contents such as 3D movies have received a lot of attention from the public. Humans watch videos using their eyes, and both eyes receive images which have two different viewpoints. We generally call those two images stereo images. Stereo images give a 3D effect to human eyes and this characteristic makes images be more realistic. Not only using stereo images but also using multi-view images which have more than three different viewpoints also put the 3D effect on images. We can get multi-view images using a lot of cameras. However, these camera systems cannot be located in a small area. In order to overcome this constraint, a view synthesis method was proposed. The view synthesis method makes virtual viewpoints using several color images and depth information. We can obtain depth information using stereo matching methods.

Stereo images are captured by stereo cameras and they have disparity information of each viewpoint. An object which is located near cameras has a large disparity value. On the other hand, the object which is located far from cameras has a small disparity value. Stereo matching methods use this characteristic to find disparity values which represent depth information. Stereo matching methods are divided into two ways. One is a local stereo matching method and the other is a global stereo matching method. The local stereo matching method finds the disparity values between stereo images using similarity measurement. It calculates similarity cost between left and right images. This measurement has several ways to obtain the matching cost such as sum of absolute differences (SAD), sum of squared differences (SSD) and normalized cross correlation (NCC).

The global stereo matching method uses the Markov random fields (MRF) model to define an energy function. The energy function is composed of data term and smoothness term. The data term represents how correct the disparity value of current pixel is. This term is very similar with cost function in the local stereo matching. The smoothness term checks the continuities between the current pixel and neighbor pixels. Since it considers neighbor pixels' disparity values, the global stereo matching method usually has more accurate results in disparity maps comparing with the local stereo matching method. Once the energy function is defined, an optimization process will be needed to find the optimal disparity value which minimizes the energy function. There are many ways to optimize the energy function such as belief propagation (BP), graph cuts (GC), dynamic programming (DP), and so on [1-3].

One of important issues in stereo matching methods is an edge preserving problem in the disparity map. In the local stereo matching method, if we use the small sized block, we can get a well edge preserved result. However, the matching block is smaller, the result includes much noise. On the other hand, if the block size is large, then edge regions in the disparity map are not preserved well. The global stereo matching method also has the edge preserving problem. The smoothness term in the energy function sometimes has a bad effect on the edge preserving in the disparity map. Because it considers neighbor pixels, it may determine the wrong disparity value in edge areas. In this paper, we introduce a conventional edge preserving disparity estimation using a distance transform in the global stereo matching method. After that, we show our proposed method which modifies the distance transform to reduce the implementation time with a small loss in bad pixel rates (BPR) comparing to the conventional method.

II. EDGE PRESERVING USING DISTANCE TRANSFORM

A. Distance Transform

The distance transform determines weight values corresponding to the distance from the edge. In order to use this transform, we have to detect edge regions in the image. An edge detected image has binary values; edge regions are 255 and the other regions are 0. Since the purpose of distance transform is weighting to non-edge regions, we change the value of edge regions from 255 to 0 and the other regions from 0 to 255. After that, a 3x3 kernel is applied to the edge

978-1-4799-8641-5/15/$31.00 ©2015 IEEE

image. This kernel transforms the edge image to the distance transformed map (DT map) [4].

Eq. 1 shows the kernel of distance transform. In this equation, $r^{k}_{i,j}$ is a pixel value in the edge image. i and j represent a pixel position, k is a number of iteration. We can define the weight of current pixel by checking a minimum value in the kernel. α and β are the strength of distance transform.

$$r^{k}_{i,j}=\min\begin{bmatrix} r^{k-1}_{i-1,j-1}+\beta & r^{k-1}_{i,j-1}+\alpha & r^{k-1}_{i+1,j-1}+\beta \\ r^{k-1}_{i-1,j}+\alpha & r^{k-1}_{i,j} & r^{k-1}_{i+1,j}+\alpha \\ r^{k-1}_{i-1,j+1}+\beta & r^{k-1}_{i,j+1}+\alpha & r^{k-1}_{i+1,j+1}+\beta \end{bmatrix} \tag{1}$$

B. Definition of Weight Functions

Weight functions are defined by two different components. First weight function is calculated by the DT map and second one is determined by the color image [4]. Based on the DT map, non-edge regions are weighted more than edge regions. By emphasizing non-edge regions, disparity values around edges can be determined more accurately. The weight function is shown as follows.

$$f(dt_t)=1\text{-}\exp\left(-\frac{dt_t^2}{2\sigma_f}\right) \tag{2}$$

In equation (2), t is the pixel position in the block and dt_t is the pixel value at t.

The color image also gives information of weight function. As the current pixel has a large probability that the current pixel value is similar with neighbor pixels'. The weight function of color image can be defined in equation (3).

$$g(|I_{L,s}\text{-}I_{L,t}|)=\exp\left(-\frac{|I_{L,s}\text{-}I_{L,t}|^2}{2\sigma_g}\right) \tag{3}$$

In (3), the color weight function g is also calculated by same block that is used in the DT map. s is the current pixel and I_L represents a luminance of current pixel. This function gives the large weight to pixels which have small color differences. Those two weight functions are applied to the data term to calculate the energy function.

$$E(d_s) = \sum_s D_s(d_s) + \sum_{s,t\in N(s)} S_{s,t}(d_s, d_t) \tag{4}$$

$$D_S(d_s) = \frac{\sum_{t\in N(s)} W_{s,t}(dt_t)\cdot F_{s,t}(d_s)}{\sum_{t\in N(s)} W_{s,t}(dt_t)} \tag{4}$$

The energy function is shown in equation (4). D_s is the data term and $S_{s,t}$ is the smoothness term. In this edge preserving algorithm, the data term is modified to preserve edge regions. The data term which is calculated by a weight function $W_{s,t}$ is represented as equation (5). In (5), $W_{s,t}$ is the multiple of two weight functions; (2) and (3). $F_{s,t}$ is the absolute differences between the left block and the right block.

III. Disparity Estimation Using Modified Distance Transform

A. Problems of Conventional Distance Transform

The conventional distance transform calculates the transform iteratively and it causes the increase of computation complexity. In order to reduce the complexity, we can limit the number of iteration transform. However, limiting the iteration number has some constraints at top horizontal lines in the transformed image. Fig. 1 shows results of 1st iteration distance transform.

In Fig. 1, red boxes represent non-transformed pixels. If we implement 1st iteration transform, we cannot get enough information to preserve edge regions. Therefore, the conventional distance transform should be operated iteratively. In this paper, we propose a modified distance transform to improve this problem.

255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
9	9	9	9	9	9	10	19
0	0	0	0	0	0	9	18
9	9	9	9	9	0	9	18
18	18	18	18	9	0	9	18
27	27	27	27	9	0	9	9
36	36	36	36	10	9	0	0

Fig. 1. First Iteration Result

B. Modified Distance Transform

The modified distance transform (MDT) has the same pre-processing to get an initial edge image as the conventional distance transform. This transform uses a window kernel and the kernel size can be determined by users. Fig. 2 describes an example of MDT.

In Fig. 2, the 3x3 window kernel is used for MDT and we use the Canny edge detection to obtain the edge image [5]. If the kernel includes the edge, the MDT calculates city-block distances from edge regions to do the transform. If multiple edges are included in the window kernel, we calculate the distance of each pixel using the nearest edges. Eq. 6 is the equation of city-block distance.

$$D(s,t) = (|x-s| + |y-t|)\cdot 10 \tag{5}$$

In equation (6), x and y are pixel positions that we have to calculate. s and t are edge positions in the kernel. We can get an MDT map by using this equation.

In order to reduce much implementation time, we use a pixel by pixel cost function at non-weight regions which have 255 pixel values in the MDT map. A new data term can be defined as follows.

$$D_s(d_s) = \begin{cases} \frac{\sum_{t \in N(s)} W_{s,t}(dt_t') \cdot F_{s,t}(d_s)}{\sum_{t \in N(s)} W_{s,t}(dt_t')}, & if\ dt_t' \neq 255 \\ |I_L(x_s - y_s) - I_R(x_t + d_s, y_t)|, & otherwise \end{cases} \quad (6)$$

In equation (7), $W_{s,t}$ uses the value dt_t' which comes from the MDT map. If there are all 255 values in the kernel, then we consider that there are no edges. In this case, we apply the pixel by pixel cost function. On the other hand, if there are values which are not 255, then we consider that there are some edges in the kernel. So, we use the same data term as the conventional distance transform.

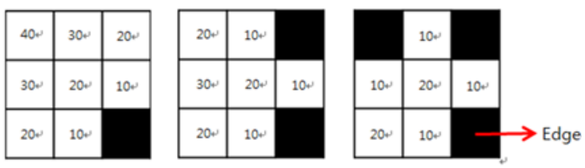


Fig. 2. Example of MDT

C. Occlusion Handling Method

An occlusion problem is also one of the important issues in the stereo matching. In this paper, we compensate occlusion regions in the disparity map which is the result of edge preserving stereo matching. In order to handle occlusion regions, we have to detect that region first. Occlusion regions can be detected by using a cross check method. The cross check method uses two disparity maps which have different viewpoints. Fig. 3 represents how to detect occlusion regions in the left disparity map.

In Fig. 3, the current pixel (x, y) in the left image has the disparity value d and then we can find the correspondent point $(x\text{-}d, y)$ in the right image. If the disparity value in the right image is not same with the disparity value of current pixel, then the pixel (x, y) is considered as an occlusion region. In the same way, we can detect occlusion regions of left disparity map. Occlusion regions are determine in consideration of ± 1 error.

Finally, we can handle occlusion regions using the disparity value of neighbor pixels. Occlusion regions in the left disparity map are usually detected on the left side of objects. Therefore, we handle those regions using neighbor pixels which are located in the left side of them.

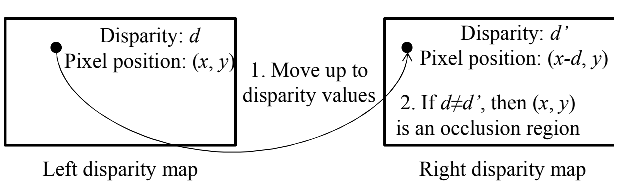


Fig. 3. Cross Check Method

IV. Experiment Results

A. Results of MDT Algorithm

We applied MDT algorithm to four test images: *Teddy*, *Cones*, *Tsukuba* and *Venus*. In order to implement the MDT, we use the Canny edge detection as the pre-processing and the 3x3 window kernel is used for calculating the weight function. We used hierarchical belief propagation (HBP) to find the optimal disparity value in the global stereo matching method [6].

Fig. 4 shows the result images of DT and MDT. Fig. 4(b) is the result of global stereo matching using only HBP, Fig. 4(c) is the result of DT and Fig. 4(d) is the result of MDT. DT and MDT algorithms do not have exceptional results when we see with naked eyes. Table 1 is a comparison table and it shows more objective values among those results.

Table 1 shows that the proposed method has better discontinuity BPR values than the original global stereo matching algorithm using HBP. Also, it has lower total BPR values than HBP algorithm. The conventional DT algorithm also has better BPR result than original HBP. However, our proposed algorithm has the best result in this experiment. The result of implementation time is also described as follows.

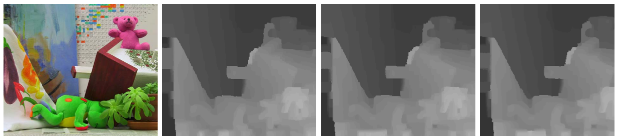
Table 2 represents the implementation time of DT and MDT algorithms. In Table 2, the proposed method reduces the complexity about 21.28%.

TABLE I. Comparison of Bad Pixel Rates

	BPR (%)					
	HBP		***HBP-DT***		***HBP-MDT (Proposed)***	
	BPR all	***BPR disc.***	***BPR all***	***BPR disc.***	***BPR all***	***BPR disc.***
Teddy	17.24	23.27	16.93	22.76	16.94	22.49
Cones	15.61	17.32	15.78	16.47	15.72	16.69
Tsukuba	4.40	14.14	3.99	10.32	3.99	10.36
Venus	2.69	12.37	2.90	16.16	2.78	14.35
Avg	9.99	16.78	9.9	16.43	9.86	15.97

TABLE II. Implementation Time Comparison

	Time (sec.)	
	HBP-DT	***HBP-MDT (Proposed)***
Teddy	126.28	90.47
Cones	124.40	113.34
Tsukuba	21.70	15.88
Venus	40.17	26.33
Average	78.14	61.51



(a) Original Image (b) HBP (c) HBP-DT (d) HBP-MDT

Fig. 4. Result Images

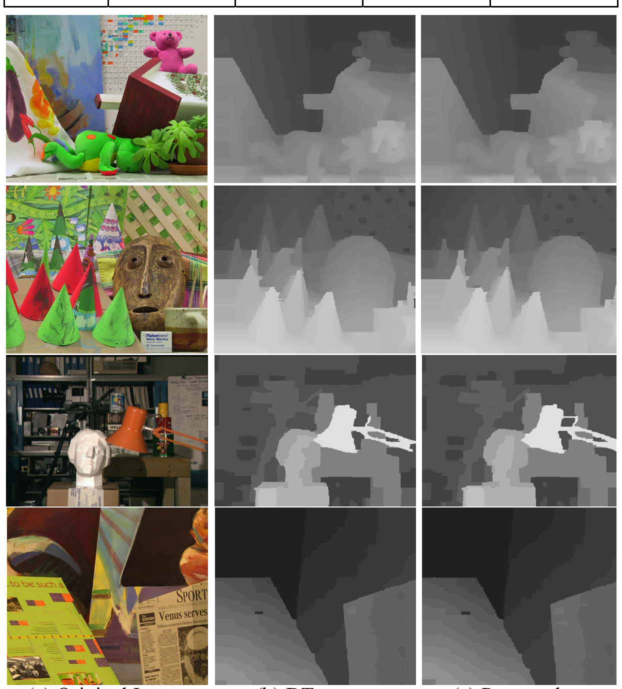
B. *Results of Occlusion Handling*

We applied the occlusion handling to proposed algorithm and compared with the result of conventional DT which is also occlusion handled. The BPR results of both algorithms are shown as Table 3. In Table 3, the results show that our proposed algorithm has 0.2% loss in the total BPR and also has 0.29% loss in the discontinuity BPR.

Fig. 5 is the results of occlusion handling. It is easy to check that occlusion handled images have better quality than the results without occlusion handling. Fig. 5(a) is the original image, Fig. 5(b) is the result of conventional DT which is applied occlusion handling and Fig. 5(c) represents the proposed method using occlusion handling.

TABLE III. BAD PIXEL RATES OF OCCLUSION HANDLING

	BPR (%)			
	DT-occ		***MDT-occ (Proposed)***	
	BPR_all	***BPR_disc.***	***BPR_all***	***BPR_disc.***
Teddy	13.83	21.78	13.70	22.23
Cones	13.08	16.38	13.26	16.68
Tsukuba	2.74	11.60	3.47	13.26
Venus	1.48	13.84	1.47	12.59
Average	7.78	15.9	7.98	16.19



(a) Original Image (b) DT-occ (c) Proposed

Fig. 5. Results of Occlusion Handling

V. CONCLUSIONS

In this paper, we propose the modified distance transform to improve the iterative computation of the conventional distance transform. In addition, we detect occlusion regions and handle those regions by using neighbor pixel values. From our experiments, the modified distance transform preserves edge regions more accurately than the conventional DT algorithm. When we apply the occlusion handling to both results, proposed algorithm has 0.29% loss in the BPR of discontinuity and 0.2% loss in the total BPR. However, the complexity of proposed method is 21.28% lower than the conventional DT algorithm.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(No. 2011-0030079)

REFERENCES

[1] J. Sun,. N.N. Zheng, and H.Y. Shum, “Stereo Matching Using Belief Propagation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, July 2003.

[2] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[3] O. Veksler, “Stereo Correspondence by Dynamic Programming on A Tree,” IEEE Conference on Computer Vision and Pattern Recognition, vol 2, pp. 384-390, June 2005.

[4] W.S. Jang, and Y.S. Ho, “Discontinuity Preserving Disparity Estimation with Occlusion Handling,” Journal of Visual Communication and Image Representation, vol. 25, no. 7, pp. 1595-1603, Oct. 2014.

[5] J. Canny, “A Computational Approach to Edge Detection,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-8, Issue 6, pp. 679-698, Nov. 1896.

[6] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, “Real-time Global Stereo Matching Using Hierarchical Belief Propagation,” BMVC, vol. 6, pp. 989-998, Sep. 2006.