Flickering Elimination Method for Background Region in Depth Video

Dong-Won Shin and Yo-Sung Ho
School of Information and Communications
Gwangju Institute of Science and Technology (GIST)
123 Cheomdan-gwagi-ro, Buk-gu, Gwangju 500-712, Korea
Email:{dongwonshin, hoyo}@gist.ac.kr

Abstract— **In three-dimensional (3D) video systems, the depth image is very important to represent complete 3D contents. However, there is a flickering problem in the original depth video in the temporal domain. It is quite annoying to users. In this paper, we propose a flickering elimination method in the original depth video from a depth camera. In order to solve this problem, first we use a depth weighted joint bilateral filter to fill hole areas in the original depth image. Then, we employ a temporal mean filter to eliminate a flickering effect on the refined depth video and classify static and moving areas. Depending on the type of areas, we adaptively choose the refined depth image or temporal mean filtered depth image. Experimental results show that the proposed method reduces flickering effects significantly in the background of depth video.**

Keywords—flickering effect; depth image; temporal mean filter;

I. Introduction

Recently, we can easily access many products which are combined with three-dimensional (3D) image technique as customer's interest for 3D image is growing. An environment that whoever can easily see 3D contents is provided as building some base structures for the 3D movie to the conventional movie theater or through a lot of 3DTV released in the market. By using this, we can experience a realistic feeling and depth perception that does not exist in 2D images.

The sense of depth can be derived from a depth image obtained through various depth acquisition methods. We can acquire the depth image by using active and passive methods: the active method employs a depth camera to get 3D information and the passive method applies a pattern matching to a stereo color image to obtain 3D information [1]. If depth images do not accurately obtain in the depth image acquisition stage, some noises generated from the acquisition step are propagated to user end and they can feel uncomfortable sense on the 3D contents. So, the depth images play an important role throughout the overall 3D image system [2].

However, the original depth image obtained from the depth camera shows an unstable appearance when we observe it in terms of the temporal domain. That is, it shows a flickering showing a huge depth difference depending on the temporal domain. It is generally shown in the active depth acquisition method which uses depth cameras [3]. Fig. 1 shows the example of the flickering from Microsoft Kinect depth camera as 30 frame per second (fps). We can find the inconsistent flickering appearance usually along with the boundary of the object such as a bookshelf, chair and human. Conclusively, this kind of the flickering effect gives users dizzy and uncomfortable feelings to enjoy the 3D contents.

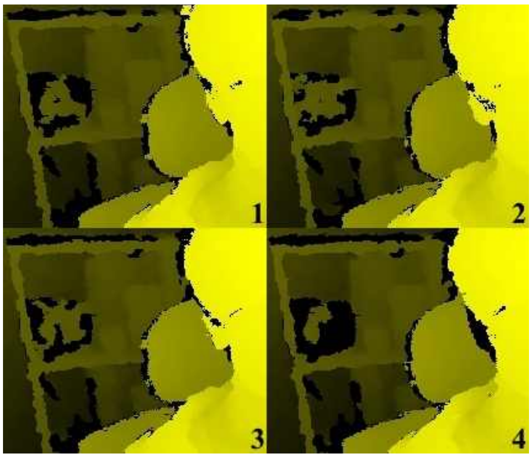

Fig. 1. Original depth video

II. Proposed Method of Flickering Elimination

Fig. 2 shows the flowchart of proposed method. First, we perform a depth weighted joint bilateral filter to fill hole area which has no depth value. Next, we store refined depth images into an array and apply a temporal mean filter on it. However at this point, when we perform the temporal mean filter on the temporal domain, we can eliminate the flickering but there still exist a problem on a residual image. In order to eliminate the residual image problem derived from temporal mean filter, we classify parts of the depth image into two category; static areas and moving areas. And then, we apply the temporal mean image on the static areas and the refined depth image on the moving area. After this step, we can get a final depth image.

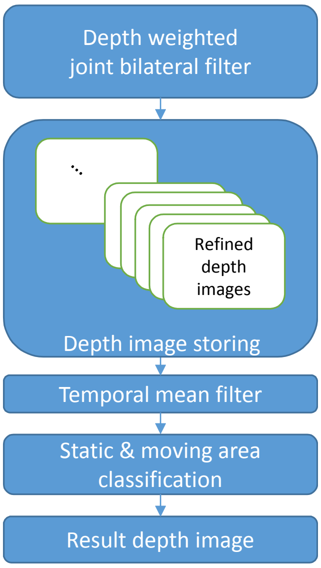

Fig. 2. Flowchart of the proposed method

A. *Depth weighted joint bilateral filter*

In this paper, we propose a depth weighted joint bilateral filter in order to fill hole areas which has no depth value in the original depth image. Equation (1) represents a structure of the propose method.

$$D_o(x,y)=\frac{\sum\limits_{u\in\vec{u}_p}\sum\limits_{v\in\vec{v}_p} W(u,v)\cdot D_i(u,v)}{\sum\limits_{u\in\vec{u}_p}\sum\limits_{v\in\vec{v}_p} W(u,v)} \qquad (1)$$

D_o stands for an output depth pixel and D_i stands for an input depth pixel. *(x, y)* and *(u, v)* represent a center pixel position and a neighbor pixel position of the filter kernel respectively. Vector $\vec{V}_p$ and vector $\vec{U}_p$ mean a set for vertical and horizontal position in the filter kernel. Lastly, *W* stands for a weighting function. Fig. 3 shows the meanings of the each symbol graphically. The original joint bilateral filter is consist of a multiplication with two Gaussian function. This function is a bell shaped function which has the highest weighting at the mean value position, so it has a low weighting value if the position is going far from the center.

In the proposed method, an additional Gaussian function reflecting a depth difference between center and neighbor pixels is appended to it. Equation (2) shows a proposed weighting function *W*.

$$W(u,v)=\begin{cases} 0 & D_i(u,v)=0 \\ g(u,v)\cdot f(u,v)\cdot d(u,v) & otherwise \end{cases} \qquad (2)$$

g(u, v) and *f(u, v)* are weighting factors reflecting a color difference and a range difference between center and neighbor pixels respectively. So far, it is the weighting factor for conventional joint bilateral filter [4]. We add a weighting factor *d(u, v)* reflecting a depth difference between center and neighbor pixels additionally. Fig. 4 shows an example for a ground truth depth image. In Fig. 4, a red box means a filter kernel and a red dot at the center represents a center pixel in the filter kernel. A blue dot means neighbor pixel. It is representing depth values by a gray level color intensity so we can know that there is a big difference between center and neighbor pixel in terms of the depth value. In this case, an effect on an output depth value is needed to be low about a neighbor pixel. So we need to adaptively apply a weighting depending on it. Equation (3) represents a structure of a weighting factor *d(u, v)*.

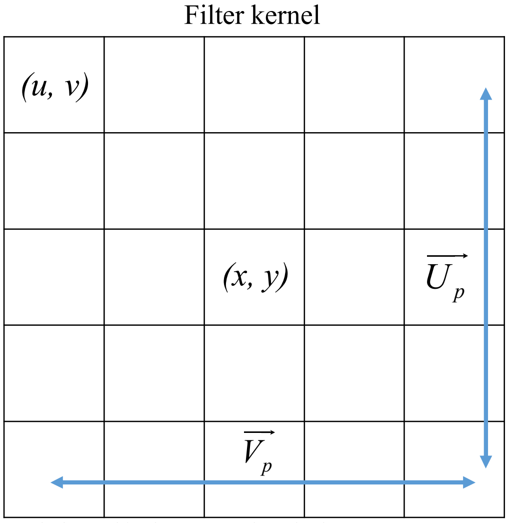

Fig. 3. Symbols used in the proposed method

$$d(u,v)=\exp\left\{-\frac{\left|D_i(x,y)-D_i(u,v)\right|^2}{2\sigma^2}\right\} \qquad (3)$$

By using this weighting factor *d(u, v)*, we can clear the object boundary region more than an original joint bilateral filter and remove intermediate noises in the discontinuity region.

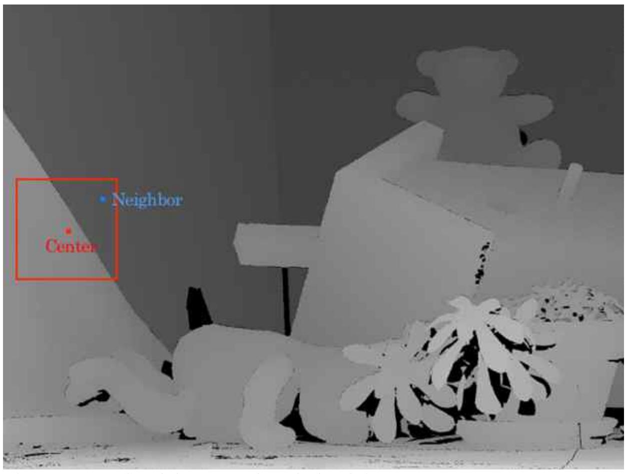

Fig. 4. Example of the ground truth depth image

B. *Temporal mean filter*

Temporal mean filter is a filter that is calculating a mean value on a temporal array of the depth image. Equation (4) represents a structure for the temporal mean filter.

$$D_M(x,y) = \frac{\sum_{t=1}^{N} D_o^t(x,y)}{N} \qquad (4)$$

$D_o^t(x,y)$ stands for a depth pixel at *(x, y)* position on *t* frame and *N* means the number of depth images in the image array. That is, (4) shows to obtain a mean value of depth values at *(x, y)* position in all depth images stored on a depth array. We can eliminate the flickering event by using this equation but we need to consider a residual image which a previous depth frame affects a next depth frame. In order to solve this problem, we classify the depth image into static and moving areas and adaptively substitute the depth values depending on the type of the area.

C. *classification of the static and moving area*

Mainly, a static area means a background area which has no movement. A moving area means a region that has a lot of movement such as a foreground region and an area which objects are moving in the foreground region. Fig. 5 shows the flowchart for the area classification.

First, we separate the foreground and background area by using the temporal mean depth image obtained from Section 2.1. The separation of the foreground and background is operated by taking a threshold on the depth image; under the threshold is classified to the background and over the threshold is classified to the foreground. Next, we can extract moving areas by applying (5).

$$D_{MT}(x,y) = \max_{0\le t\le N}(D_o^t(x,y)) \qquad (5)$$

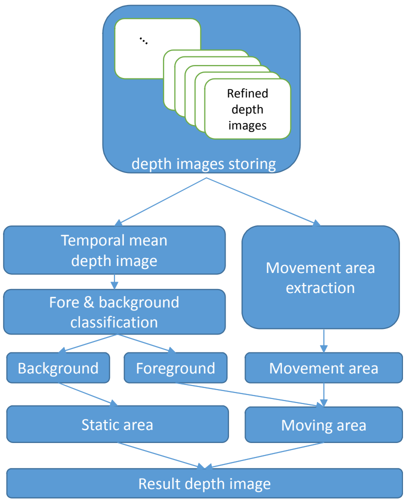

Fig. 5. Flowchart for the area classification

This equation takes the maximum value at *(x, y)* position. By using this equation, I shows a figure representing extracted moving area in Fig. 6. The red stroked region represent moving areas.

Finally, if a depth pixel position is on the static area, we can substitute the pixel by using the temporal mean image. If it is on the moving area, we can change the pixel by using the refined depth pixel. Conclusively, we adaptively used the temporal mean image and refined depth image depending on the type of the region.

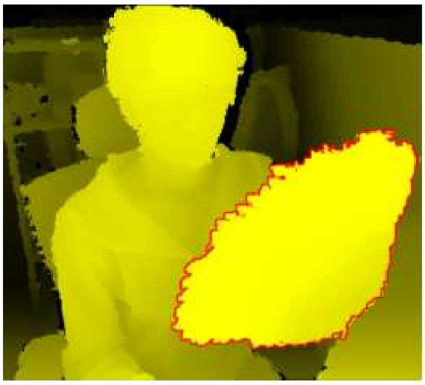

Fig. 6. Extraction of moving area

Finally, if a depth pixel position is on the static area, we can substitute the pixel by using the temporal mean image. If it is on the moving area, we can change the pixel by using the refined depth pixel. Conclusively, we adaptively used the temporal mean image and refined depth image depending on the type of the region.

III. Experimental Results

In this experiment, Microsoft Kinect was used and the size of depth image is 640×480. We used a GPU parallel programming to accelerate a calculating speed and the GPU model is Nvidia Geforce GTX Titan [5]. Fig. 7 shows the result depth video. In this figure, we can know that the flickering is dramatically reduced on the background area.

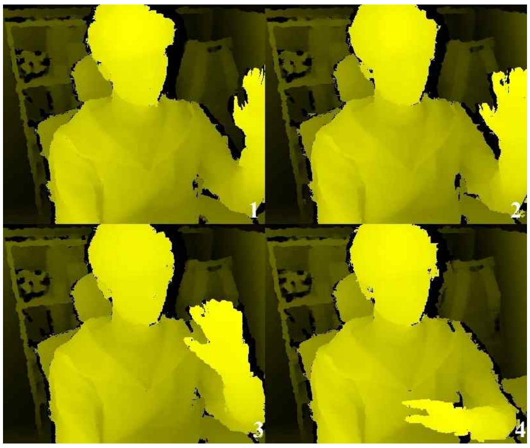

Fig. 7. Result of depth video

For the qualitative analysis, we represents the change of the depth pixel value in Fig. 8, Fig. 9 and Fig. 10. The horizontal axis means frame numbers along with the time and the vertical axis means depth values. The blue curve represents the depth value change in the original depth video and orange color curve represents the depth value change in the result depth video. Following figures show a trend for the depth value obtained from a boundary region. Through the result graph, we can see that the depth value of the original depth video vibrates excessively while the depth value of the result depth video is relatively stable.

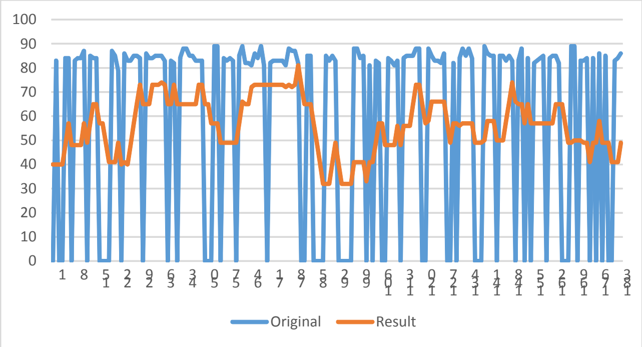

Fig. 8. Change for depth values on the boundary region 1

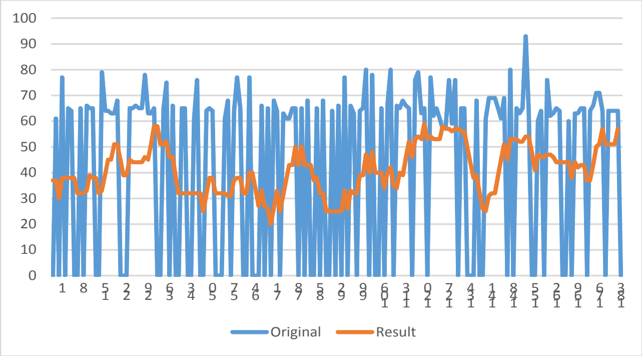

Fig. 9. Change for depth values on the boundary region 2

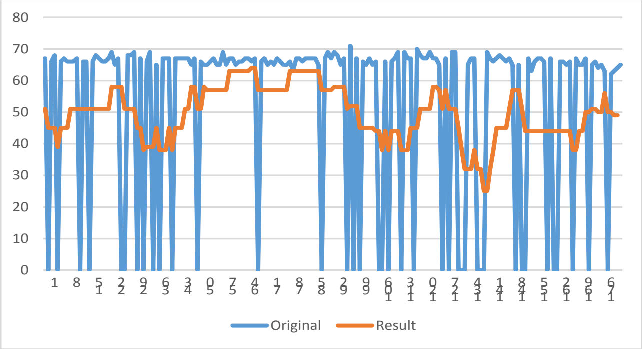

Fig. 10. Change for depth values on the boundary region 3

We show the variance value for each boundary region in Table. 1. We can see that it has lower variance values in case of the result depth images than original depth images.

TABLE 1. Variance of the boundary regions

Variance	Original	Result
Boundary region 1	1538.485	131.9571
Boundary region 2	1052.025	84.21052
Boundary region 3	749.556	70.80516

IV. Conclusion

In this paper, we discussed a method improving a temporal consistency in the background region of the depth video obtained from a depth camera. After we employ a depth weighted joint bilateral filter to eliminate hole area in the original depth video, we apply a temporal mean filter to eliminate the flickering. In order to eliminate a residual effect, we classify the depth image into static and moving areas and adaptively select the type of the depth value. We have shown that the proposed method reduces the flickering effect in the background region and improves the temporal consistency of the depth image.

Acknowledgment

This work was supported by the ICT R&D program of MSIP/IITP.[11-921-05-001, Development of Immersive Smart-work Core Technology for Collaboration among Multiple Parties]

References

[1] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, “Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability,” Image Commun., vol. 22, no. 2, pp. 217-234, Feb. 2007.

[2] C. Fehn, “Depth-image-based rendering (DIBR), Compression and Transmission for a New Approach on 3-D TV,” Proc. of SPIE Conference Stereoscopic Displays and Virtual Reality Systems, vol. 5291, pp. 93-104, Jan. 2004.

[3] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, “Temporal filtering for depth maps generated by Kinect depth camera,” 2011 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON 2011), pp. 1-4. May 2011.

[4] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” ACM Transactions on Graphics, vol. 26, no. 3, pp. 1-5, July 2007.

[5] D. Shin and Y. Ho, “Real-time Depth Image Refinement using Joint Bilateral Filter,” Proc. of 2013 Korean Society of Broadcast Engineers Autumn Conference, vol. 19, pp. 116-119, Nov. 2013.