Time-of-Flight Image Enhancement for Depth Map Generation

Yunseok Song and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
123 Cheomdangwagi-ro Buk-gu, Gwangju 61005, Republic of Korea
E-mail: {ysong, hoyo}@gist.ac.kr

***Abstract*—Time-of-Flight (ToF) cameras are easily accessible in this era. They capture real distances of objects in a controlled environment. Yet, the ToF image may include disconnected boundaries between objects. In addition, certain objects are not capable of reflecting the infrared ray such as black hair. Such problems are caused by the physics of ToF. This paper proposes a method to compensate such errors by replacing them with reasonable distance data. The proposed method employs object boundary filtering, outlier elimination and iterative min/max averaging. After acquiring the enhanced ToF image, this can be applied to depth map generation by using the ToF camera with other color cameras. The experiment results show improved ToF images which lead to more accurate depth maps.**

I. Introduction

Depth maps represent the geometry of a scene, containing camera-to-object distance data. The accuracy of depth maps is crucial in virtual view synthesis and 3D modeling [1]. Common depth map generation techniques are based on either stereo matching or active depth sensing. The former method finds correspondences between stereo images and converts them to depth data. Yet, this process can be very time consuming depending on the disparity range [2-4]. In addition, correspondence matching is problematic in textureless regions. The latter method exploits depth sensors. Time-of-Flight (ToF) and structured light sensing are popular techniques for depth sensors. Due to the physics of the sensor, only low resolution images can be captured. Moreover, since the capture range is limited, this method cannot be used in outdoor environment.

This paper focuses on quality improvement of captured ToF sensor images. Mesa Imaging SR4000 [5] ToF sensor is used for experiments in this paper. In regards to this ToF sensor, the capture range is 5 m and the resolution is 176×144. The ToF is measured as the object reflects the infrared ray sent from the ToF sensor. However, certain black objects may absorb the ray which creates inaccurate distance data in the ToF image. Such examples include human hair, clothes and rear parts of computer monitors. In addition, the boundary between the object and the background is not correctly represented. In particular, a distance value which lies in the range of object distance and the background distance would be stored; yet in reality, nothing exists at such a distance. ToF images are often employed in 3D image/video applications. The errors in the raw ToF image may lead to quality degradation of the multimedia application.

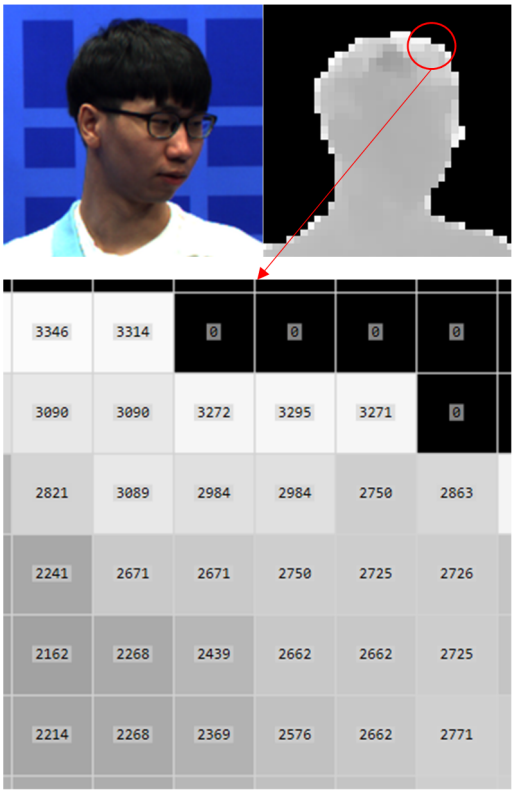


Fig. 1. Inaccurate distance data within a foreground object.

Fig. 1 is an example of inaccurate data in the ToF image. Each pixel contains the camera-to-object distance in millimeters, ranging from zero to 5,000; thus, the ToF image is represented by 16 bits. In this paper, the background is represented by zero (black) for display purpose. In addition, 16-bit distance values are displayed with maximized contrast.

In the enlarged region of Fig. 1, the object distance is inconsistent. For example, there is a sudden increase from 2,241 to 2,821 even though they are right next to each other. In addition, values such as 3,314 and 3,346 are neither foreground nor background since the background distance is 3.5 m and the maximum distance of the object is 3.1 m. The proposed method compensates such irregular distance values using neighbor information.

II. Proposed Method

The proposed method comprises three steps: object boundary filtering, iterative outlier elimination and iterative min/max averaging. The capturing environment is an indoor studio with chromakey background. The minimum and maximum distances of the foreground object is inputted by the user manually. Since the distance of the background is known, the proposed method is able to modify only the foreground region. In the following figures, the background region is represented by zero (black) for display purpose of 16-bit ToF images.

A. Object Boundary Filtering

In the first step of the proposed method, the boundary values between the foreground and background are modified. Filtering is performed on both horizontal and vertical directions. First, using a 1-tap 3-pixel wide window, two criteria are checked: whether 1) the current pixel is background, 2) one of its neighbors is background. If these two are satisfied, this pixel is determined to be a boundary region value; object boundary filtering is applied to this pixel. The maximum distance value of its two neighbors becomes the new center pixel value. The procedure for object boundary filtering in the horizontal direction is depicted in Fig. 2. The same process is executed for the vertical direction as well.

Fig. 3 exhibits the effect of object boundary filtering. The result image shows that irregular boundary values (red circles) are replaced with neighbor values that are valid. The values within the object are not altered. Table 1 shows the pseudo code for the process of object boundary filtering.

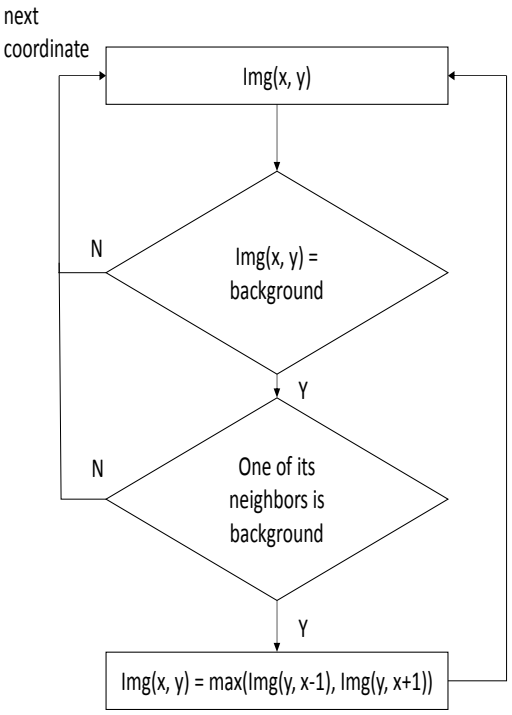


Fig. 2. Flowchart of object boundary filtering in the horizontal direction.

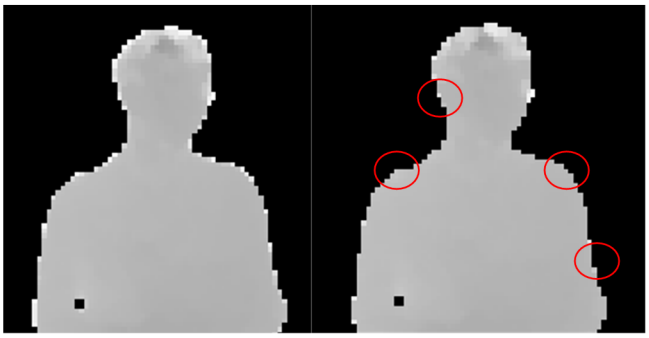
TABLE I. Pseudo Code for Object Boundary Filtering

```
Boundary_filtering(img)
for (y = 1; y < height – 1; y++)
{
   for (x = 1; x < width – 1; y++)
   {
      // BG: background value

      // Horizontal filtering
      if (img(y, x) != BG &&
    (img(y, x – 1) == BG || img(y, x+ 1) == BG) )
         img(y, x) = MAX(img(y, x - 1), img(y, x + 1)

      // Vertical filtering
      if (img(y, x) != BG &&
    (img(y - 1, x) == BG || img(y + 1, x) == BG) )
         img(y, x) = MAX(img(y - 1, x), img(y + 1, x)

   }
}
```



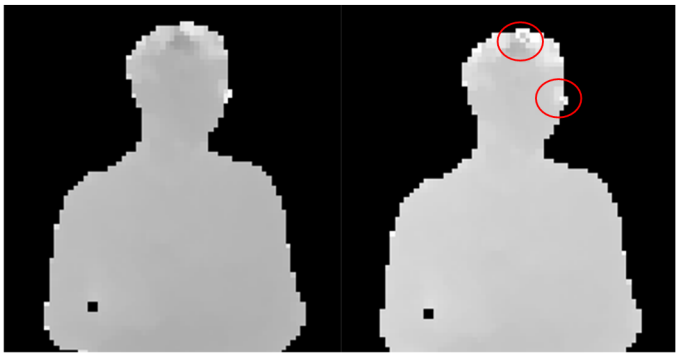
(a) Before boundary filtering (b) After boundary filtering

Fig. 3. Result of object boundary filtering.

B. Iterative Outlier Elimination

After the first step, values which should be regarded as outliers still exist within the foreground object. They possess data distance data in between the maximum object distance and the background distance. Furthermore, distances closer than the minimum object distance are invalid. These outliers are caused by inaccurate depth sensing. By physics, black objects tend to absorb the infrared ray, hence, erroneous distance data are stored within the object.

As the second step, the proposed method iteratively eliminates such outliers by replacing them with neighbors that possess valid distance data. A 3×3 window is used for searching valid neighbors. Iterations are necessary since all outliers cannot be removed in a single execution. The iteration ends when all the pixels in the entire object possess valid distance data. A larger window will remove more outliers but the blurring effect will increase as well. Fig. 4 displays the effect of outlier elimination. As a result of outlier elimination, the maximized contrast effect is reduced, i.e. all the pixel values in the image lies within the range of object distance. Table 2 shows the pseudo code of iterative outlier elimination process. The filter is designed to not let the outcome of the replacement value be the background value.



(a) Before outlier elimination (b) After outlier elimination

Fig. 4. Result of iterative outlier elimination using minimum neighbor.

TABLE II. PSEUDO CODE FOR ITERATIVE OUTLIER ELIMINATION

```
iterative_outlier_elimination(img, radius)
continue_iteration = true;

LOOP:do
{
   img = elimination_using_minNeighbor(img,
radius);
   for (y = 1; y < height – 1; y++)
   {
     for (x = 1; x < width – 1; y++)
     {
         if (img(y, x) > MAX_DISTANCE)
            goto LOOP;
     }
   }
   continue_iteration = false
} while (continue_iteration)
```

```
elimination_using_minNeighbor(img, radius)
neighbor = zeros(2×radius + 1, 2×radius + 1);

for (y = 1; y < height – 1; y++)
{
   for (x = 1; x < width – 1; y++)
   {
        if (img(y, x) == BG)
            continue
        if (img(y, x) > MAX_DISTANCE ||
            img(y, x) < MIN_DISTANCE)
        {
           for (yy = y - radius; yy < y + radius; yy++)
           {
            for (xx = x - radius; xx < x + radius; xx++)
              {
         neighbor(yy – y + radius, xx – x + radius) =
img(yy, xx);
              }
           }
           img(y, x) =
     getMinimumNeighbor(neighbor);
        }
     }
}
```

C. *Iterative Min/Max Averaging*

In the final step, minimum and maximum neighbor filters are applied independently. Then, the average of these two images compared with the image before filtering. If the difference between them is greater than a certain threshold, this process is repeated. The difference comparison assures saturation of change. The purpose is to smooth out the abrupt changes within the object, particularly in the infrared ray absorbed region. Fig. 5 illustrates the iterative averaging of minimum and maximum neighbor filtered images.

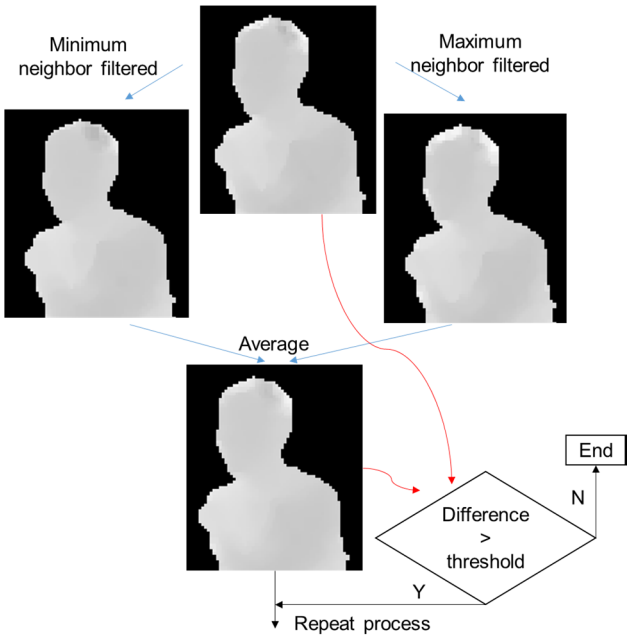


Fig. 5. Iterative averaging of minimum and maximum neighbor filtered images.

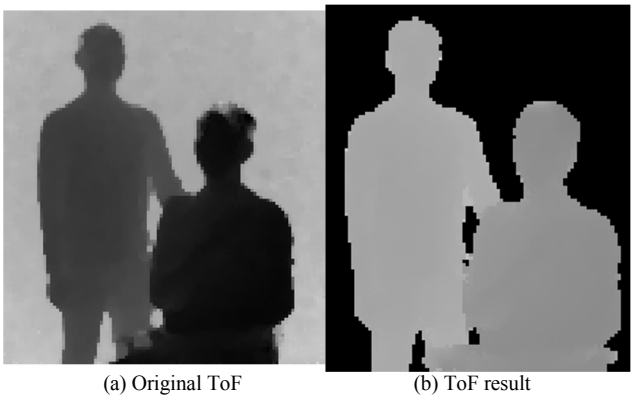
III. EXPERIMENT RESULTS

Tests were conducted on four ToF images. Further, we test the ToF image on a color camera viewpoint-depth map generation [2]. The resolution is 1280×720 for the color image and 176×144 for the ToF image. The ToF sensor captures 16-bit amplitude and confidence images, but not color images. The color image is captured by an adjacent color camera. The synchronization between cameras is guaranteed by an external sync module.

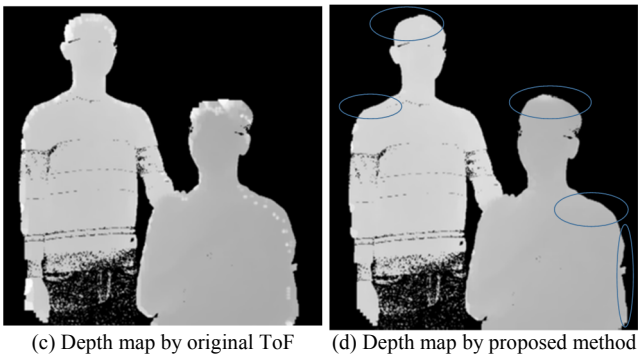
Since the original and modified Tof images are 16-bit data, they are converted to 8-bit grayscale images for display purpose. In real applications, the 16-bit data would be employed without modification. Fig. 6, Fig. 7 and Fig. 8 exhibit original color image, original ToF image and the enhanced ToF image by the proposed method. Since ground truth depth maps do not exist in these cases, objective evaluation cannot be carried out.

Subjectively, in the hair part, lost data are filled with reasonable values considering the available neighbor data. Due to the iterative averaging, the torso and arm regions are blurred as a trade-off. A more effective technique to handle this would

be investigated further by the authors. The figures show only the main object region of the 1280×720 depth map. It is confirmed that disconnected regions appears more natural in the result image.



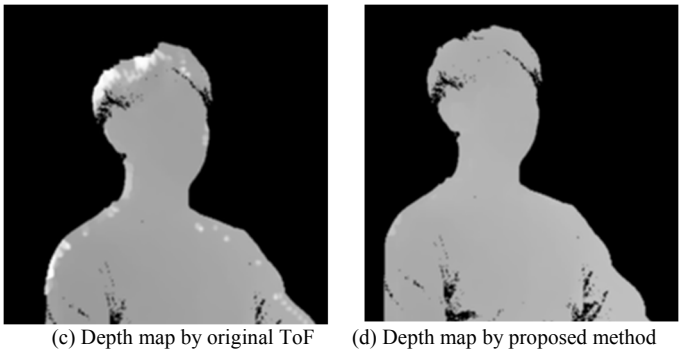
(a) Original ToF (b) ToF result



(c) Depth map by original ToF (d) Depth map by proposed method

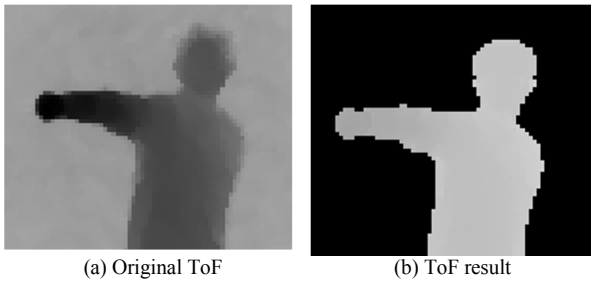
Fig. 6. Original and result image of "Test image 1".

(a) Original ToF (b) ToF result

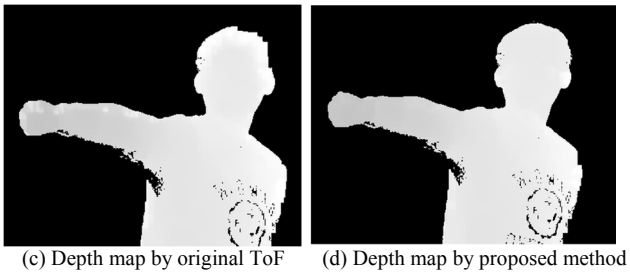


(c) Depth map by original ToF (d) Depth map by proposed method

Fig. 7. Original and result image of "Test image 2".



(a) Original ToF (b) ToF result



(c) Depth map by original ToF (d) Depth map by proposed method

Fig. 8. Original and result image of "Test image 3".

IV. Conclusion

ToF cameras are beneficial in capturing distance data. However, object-background boundary and regions that can absorb infrared rays are susceptible to inaccurate depth sensing by ToF cameras. The proposed method employs three steps to compensate such errors: object boundary filtering, iterative outlier elimination and iterative min/max averaging. Parameters including minimum and maximum object distances need to be predetermined. Using a 3×3 window, valid neighbor data are exploited for replacing and smoothing regions that are inconsistent. Since ground truth depth maps are not available, subjective quality comparison is shown. The result images appear more natural, particularly showing vast improvement in the hair region. In addition, the application of the enhanced ToF image confirms improved depth map generation when coupling it with a color camera.

Acknowledgment

This research was supported by the 'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea(ROK). [GK16C0100, Development of Interactive and Realistic Massive Giga-Content Technology]

References

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

[2] Y. Song, D.W. Shin, E. Ko, and Y.S. Ho, "Real-time Depth Generation Using Hybrid Multi-view Cameras," *Asia-Pacific Signal and Information Processing Association (APSIPA)*, pp. 1-4, Dec. 2014.

[3] Y.S. Kang, Y.S. Ho, "Disparity Map Generation for Color Image Using a TOF Depth Camera," *3DTV Conference*, pp. 1-4, May 2011.

[4] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View Generation with 3D Warping Using Depth Information for FTV," *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 65-72, Jan. 2009.

[5] Mesa SR4000 user manual, http://www.mesa-imaging.ch/