A Key Frame-based Depth Propagation for Semi-Automatic 2D-to-3D Video Conversion Using a Pair of Bilateral Filters

Jin Woo Choi, Taeg Keun Whangbo

1. First Author
Culture Technology Institute, Gachon University, cjw49@paran.com

2. Corresponding Author
Department of Interactive Media, Gachon University, tkwhangbo@gachon.ac.kr

Abstract

This study aims to propose a novel key frame-based depth propagation for semi-automatic 2D-to-3D video conversion. First, we set the depth resulted from the bilateral filtering through the previous frame color as a reference depth. We also used the color difference between the previous frame and the current frame for bilateral filtering of the depth and created the initial depth of the current frame. At all locations of initial depth, we looked for the location enough to be matched to a reference depth in order to estimate motion and did depth compensation for the previous frame to estimate the final depth of the current frame. At this time, we used the block-based matching as a matching method and applied the dynamic search area method considering the processing time. In this study, the proposed method can be used to estimate the depth as closely as possible even at the region where the values of color in object vs background are similar and the edge shapes are preserved well in the same depth as set in key frames by an operator. Our method having several actual image sequences suggested that the estimated depth map was similar to the original depth map, compared to the existing method.

Keywords: 2D-to-3D conversion, depth propagation, depth map

1. Introduction

3D stereoscopic video coding is a visual skill that makes us feel a sense of reality, for example, as if they were in the scene where images were being produced. As its application varies, it’s even recognized as the core skill for the next generation. Quite recently, 3D TVs are distributed even to our houses. In this respect, the demand for 3D contents is expected to keep increasing, but compared to this demand for 3D contents, the supply of 3D contents is still lacking. Establishing the infrastructure to create 3D contents including stereoscopic camera requires a lot of cost and efforts and thus, the conversion of the existing 2D images into 3D ones (2D-to-3D) is emerging as a good alternative. Basically, 2D-to-3D conversion consists of the processes such as interpretation of the monocular three-dimensional information contained in 2D images, creation of depth map that can be used to estimate the depth of object, and output of left and right images including binocular disparity [1-2].

Currently, 2D-to-3D conversion can be classified into three ways: full-automatic, semi-automatic, and full-manual conversion [1-3]. Full-automatic conversion, which is usually observed in real time in some 3D TVs, can convert 2D video directly to 3D one without doing any manual works. However, the three-dimensional effect of the converted 3D video is incomplete and even does not guarantee its quality. Full-manual conversion can be used to convert almost all frames manually using professional video editing software. Nevertheless, this method also has an disadvantage in that it consumes a lot of labor, time, and cost, although it’s being used to obtain high quality for 3D videos, for example, the existing 2D movie was converted to 3D one in order to re-release the movie.

Semi-automatic conversion can be used to reduce a lot of manual elements to be included in the converting process while guaranteeing the good quality of the converted 3D videos. In the same context, the object extraction/tracking having high overhead and the depth granting process being used especially in 2D-to-3D conversion with automation technology as well as user’s intervention are being studied [1-11]. Typically, the researches on the method of creating depth in all pixels within the image simply by making any marks on each object [1-4] and the method of propagating the depth of key frame being processed as precisely as possible, as shown in figure 1, into the non-key frame within the same scene automatically [1,3,5-11] are being conducted.
This study aims to improve the depth propagation using the bilateral filtering, as suggested by Varekamp et al. [5], in the key frame-based depth propagation to estimate more accurate depth map in the non-key frame. The configuration of this study is as follows: Section 2 describes the bilateral filtering-based depth propagation by presenting related works; Section 3 introduces a suggestive technique to improve this depth propagation; Section 4 presents experiment and performance evaluation; and Section 5 ends with a conclusion.

2. Related works

Bilateral filtering is an image filtering technique designed to reduce noise while preserving the original boundary [12]. For the first time, [5] proposes a method to propagate the depth of key frame into the rest frames by transforming the bilateral filtering. The method that the depth in the current frame is propagated into the next frame according to the formula (1) is introduced in [5].

$$d_{i}^{(t+1)} = \frac{\sum_{j} f_{i} w_{j}^{(t+1,1)} d_{j}^{(t)}}{\sum_{j} f_{i} w_{j}^{(t+1,1)}}$$

Herein, $d_{i}^{(t+1)}$, resulting obtained, is an estimated depth of pixel $i$ in the frame of $t+1$ and $d_{j}^{(t)}$ is the depth of pixel $j$, neighbored from the center of $i$ in the frame of $t$. $f_{i}$ outputs the value of 1 or 0 depending on the size of filter which is set as the function for spatial filtering. $w_{j}^{(t+1,1)}$ refers to color difference between pixel $i$ and pixel $j$ in the frame of $t+1$. This $w_{j}^{(t+1,1)}$ is designed to output the bigger value when the color difference gets smaller. In other words, the depth of pixel $i$ in the frame of $t+1$ is estimated by putting together the depths of $t$-framed filter size considering motion. If it’s more similar to the color of $i$, it’s propagated with higher value.

The depth propagation through bilateral filtering has an advantage in that it can continuously estimate the depth map of various non-key frames during simple implementation. On the contrary to this, it still has a problem in that the boundary is ambiguous due to the presence of similar colors despite the different depths within the filter size and that the depths can be mixed nearby the new color values in the background region newly appeared by the motion of object [5]. To solve this, [5] proposes depth compensation through motion estimation, which is a method to copy depth from the region showing the least difference through the block matching with $t$-framed depth map with no errors after dividing the $t+1$ depth map obtained as a result of bilateral filtering in the unit of block.

Lie at al. [8] proposed a trilateral filtering-based depth propagation to improve the accuracy in the block-based motion compensation as shown in [5]. This is a method to apply the trilateral filtering using the depth difference as well as the color difference in the surroundings into the weight function.
after motion compensation of the depth map in the previous map by estimating motion based on color information. If the motion vector extracted from the color information is accurate, the result of depth propagation might be satisfactory, but it’s more time consuming to obtain the motion vector. In addition, if the color values between object and background are similar, the quality of the propagated depth cannot be guaranteed due to wrong motion vector. Furthermore, it also has another problem in that as finer but blurred boundaries are increasingly accumulated during the filtering process, the depth map gets blurred if the number of the propagated frames increases.

Cao at al. [1] proposed a method to add color information-based motion vector into the spatial filter function in the existing bilateral filtering for the depth propagation of non-key frame by creating the depth map through user’s simple interaction in the key frames. In other words, it’s called as “SBF(shifted bilateral filtering),” as a method to determine neighbors depending on the motion vector of the pixel in applying the bilateral filtering. Bi-directional depth propagation having front and rear key frames is also proposed during the depth propagation process. However, it has a problem in that there are a lot of noises sounding in object boundaries as a result of the depth propagation. Thus, it seems that we need to improve it.

3. Proposed depth propagation algorithm

The existing methods are problematic in that wrong depths appeared when the colors between object and background were similar or when the occlusion area occurred, which are both limitations to the depth propagation through bilateral filtering. Our experiment on the direct implementation of the depth propagation algorithms shown in [5] and [8] compared to the existing methods we explained in Section 2 showed that the method suggested in [5] was a method to copy the similar blocks of the previous frame for motion compensation as they were, but had a problem in that the depth estimation of the detailed object motion was not well performed and that it’s not well matched between the block boundaries. In case of [8], it showed a problem that when there was a motion in the boundary where the depth difference was big, it’s blurred, and when the number of frame got accumulated, the depth map was blurred, in general. The method of [1] could not be applied into the experiment because it’s difficult to find out its specific implementation from the literature.

To solve the above problems, we devised a method to use the pair of bilateral filters. The figure 2 in the below represents the pipeline of the depth propagation.

![Figure 1. Pipeline of our depth propagation scheme](image)

First, it’s convolution-treated through the bilateral filtering, which belongs to the formula (2), using color image of the previous frame $C^{r-1}$ and depth map $D^{r-1}$, and then the reference depth $D_{\text{reference}}$ is created. At the same time, the initial depth $D_{\text{init}}$, which belongs to the formula (3), is created using the color images of previous frame $C^{r-1}$ and the current frame $C^r$ and the depth map of the previous frame $D^{r-1}$. $\alpha$ is a constant showing the color importance and thus set as 0.5 in this study.


\[ D_{\text{reference}}(x,y) = \frac{\sum_{i=-N}^{N} \sum_{j=-N}^{N} e^{-\alpha \|C(t-1) - C^i(x+i,y+j)\|} \cdot D_{t-1}(x,y)}{\sum_{i=-N}^{N} \sum_{j=-N}^{N} e^{-\alpha \|C((x,y) - C^i(x+i,y+j))\|}} \]  

(2)

\[ D_{\text{init}}(x,y) = \frac{\sum_{i=-N}^{N} \sum_{j=-N}^{N} e^{-\alpha \|C(x,y) - C^i(x+i,y+j)\|} \cdot D_{t-1}(x,y)}{\sum_{i=-N}^{N} \sum_{j=-N}^{N} e^{-\alpha \|C((x,y) - C^i(x+i,y+j))\|}} \]  

(3)

Initial depth is a result of reflecting the depth of the previous frame depending on the color similarity between the previous frame and the current frame, and has the basic shape of depth for the current frame. However, it is also found that the depth boundary is blurred instead of being preserved in the region where the colors between object and background are similar due to the limited bilateral filtering. Reference depth is the depth calculated by reflecting the color of the previous frame only or calculated by the color difference between neighboring pixels from each location of pixel. This reference depth is used as a guiding image to solve the blurring that appears more as the smaller the color difference is in the object boundary. In other words, this reference depth is the depth map created, conversely, to compensate the errors shown in the depth boundary due to similar color values between object and background during the depth propagation through bilateral filtering.

![Figure 2. Example of creating the reference depth and the initial depth in "Can" video](image)

The next processing step is to predict motion between frames by comparing the reference depth and the initial depth. When comparing the formula (2) and (3), the rest areas excluding the can in figure 3 showing no motion will have the same result as that in the reference depth and the initial depth. In the area where there is an object motion, especially in case of Can in figure 3, the direction and size of motion is to be predicted. The direction and size of motion at a certain location between frames is calculated through block matching based on the central pixels expressed in the following formula (4) and calculated at all pixel locations of the initial depth.

\[ (\mu_x, \mu_y) = \arg \min_{(\Delta x, \Delta y)} \sum_{i=-N}^{N} \sum_{j=-N}^{N} |D_{\text{init}}(x+i,y+j) - D_{\text{reference}}(v_x+i, v_y+j)| \]  

(4)

Herein, \( v_x = x + \Delta x, v_y = y + \Delta y \)

In formula (4), \( N \) is a constant that can determine the size of block for matching, and \( v_x \) and \( v_y \) are the centers of the candidate blocks that moved the current motion vectors from the centers, \( x \) and \( y \),
by as much as the displacement of expected motion, $\Delta x$ and $\Delta y$. $\Delta x$ and $\Delta y$ become, when the difference between these two blocks is the smallest, motion vectors in the cells $(\mu_x, \mu_y)$, and the final depth of the current frame is determined by formula (5).

$$D^f(x,y) = D^{f-1}(x + \mu_x, y + \mu_y)$$

During this actual implementation, the higher the size of block and the search area of candidate blocks are, the longer the processing time is. In this study, the size of block is limited to $N=2$ and the search area is made to be set dynamically to minimize the time complexity of algorithm. The way that it’s made is that the motion vector of the previous frame is remembered and then from the block of the applicable location is compared first, and then if the matching value between blocks during the searching process is smaller than the set critical value, $\epsilon$, it stops immediately. $\epsilon$ is set in a manual way as a proper value by determining the size of $N$ and the complexity of depth map, the entire object of sequence, and the degree of motion in camera.

![Motion prediction and compensation with the reference depth](image)

**Figure 3.** Motion prediction and compensation with the reference depth

Figure 4 represents the example of motion prediction and compensation using the reference depth in the video showing no camera motion. At the initial depth, the location of block, a is the region with no changes in motion, and the location of block, b is the region showing motion. The blocks matched at the reference depths of two blocks are a’, b’, respectively. Block a with no motion is discovered in the place in the same shape as block a’, and block b’ which is matched to block b is selected as the block that has the most similar shape at a location where motion is reflected. As a result of treating all pixels of the initial depth, the depth map predicted for the current frame finally is represented in the right bottom in figure 4.

4. Experimental results

This Section represents the result of testing the proposed depth propagation having actual video sequences. First, the result of testing “man” sequence (1024 x 778) [13] and “can” sequence (400 x 300) [14] that have the true depths of all frames in the first experiment was compared with the algorithm of [5]. The result was indicated in figure 5, and as the quantitative comparison, Peak-Signal-to-Noise Ratio(PSNR) with true depth of each frame expressed in formula (6) was calculated and indicated in figure 6 using a chart. In PSNR formula (6), m and n are the horizontal and vertical size in the test video, and $D_{true}$ is indicated as the true depth of the current frame and $D_{est}$ as the estimated depth by algorithm.
PSNR = 10 \cdot \log_{10} \left( \frac{255^2}{\text{MSE}} \right) 

\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \| D_{\text{true}}(i,j) - D_{\text{read}}(i,j) \|^2 

In case of “man” sequence, background is simple and object motion is small. However, the depth of key frame is generated manually and thus is a detailed image. An experiment was carried out regarding the depth propagation of 27 frames in total. The results from the two methods were all excellent. However, overall, the two results looked similar, but the proposed method suggested that it’s propagated in more detail in the boundary of object and background. PSNR, in all frames propagated, was found to be high. The average PSNR in all 27 frames was 41.89dB in [5] and 49.03dB in our method. “Can” sequence had the depth of relatively simple shape in key frames having sharp edges. It was the sequence that the object motion was big and the color having background was similar. In case of “Can” sequence, an experiment was carried out regarding the depth propagation of 20 frames in total. Overall, PSNR appeared lower than “man” video, experimented earlier. This is considered to be including the errors occurred for the reason that the changes in the entire depth could not be calculated because object moved to z-axis (Can moves rolling forward). We need to carry out further researches on this later. The average PSNR of 20 frames propagated was 25.62dB in [5] and 27.47dB in our method.

Figure 5. Result of depth propagation after +27, +20 frames, respectively, after “man” and “can” sequence

Figure 6. Comparison of PSNR with the true depth of depth propagation in “man” or “can” sequence

The next experiment was carried out using the images that have the true depths only in the key frames. Our experimental videos are ① “Island”, ② “Dice”, and ③ “HeadRotate”, which are all
promotional videos for Philips WowVx(showreel) and Heinrich-Hertz-Institute’s ④“Interview”, a famous test video. The above experimental videos are provided by Broadband Network & Digital Media Lab., Tsinghua University [15]. The experimental method is that the depth propagation is carried out from the first key frame to the second key frame, +30 frame, and PSNR is obtained on the basis of the true depth in the second key frame. In this experiment, the method in [8] as well as that in [5] is compared. Subsequently, the result is summarized in table 1 and figure 7. The experimental result suggests that the depth propagation proposed in all four images is more accurate.

Table 1. Comparison of PSNR with the true depth of depth propagation in four test sequences (over 30 frames)

<table>
<thead>
<tr>
<th>Sequence Name</th>
<th>Resolution</th>
<th>Features</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>① Island</td>
<td>960 x 540</td>
<td>texture-less regions, zoom in/out</td>
<td>23.37, 23.02, 31.85</td>
</tr>
<tr>
<td>② Dice</td>
<td>960 x 540</td>
<td>sharp-edges, simple depth, moving object, moving camera</td>
<td>20.75, 16.24, 22.07</td>
</tr>
<tr>
<td>③ HeadRotate</td>
<td>960 x 540</td>
<td>color ambiguous, moving object, elaborate depth</td>
<td>23.38, 23.48, 24.30</td>
</tr>
<tr>
<td>④ Interview</td>
<td>720 x 576</td>
<td>color ambiguous, non-rigid moving object</td>
<td>19.74, 24.92, 31.22</td>
</tr>
</tbody>
</table>

Figure 7. Chart showing PSNR in table 1

In particular, it is seen clearly that ① and ④ test sequences perform more accurate depth propagation as the proposed methods, and in case of ② and ③ test sequences, there are many occlusion areas due to moving camera and object rotation. The results appeared lower in all methods, overall. But it can be seen that our method is close but high even in PSNR, and especially if seen with naked eye, the result is good. Figure 8 shows the result of depth propagation regarding the above four test sequences. The left is the depth map propagated by as much as 15 frame and the right is the depth map propagated by as much as +30 frame where there exists the true depth enough to compare.
Figure 8. Result of depth propagation in each test video: “Island”, “Dice”, “HeadRoate”, “Interview”
Left: result of propagation over 15 frames, Right: result of propagation over 30 frames (final result)
From above: color, true depth, result of [5], result of [8], and our result

As the purpose of the proposed algorithm is not the real time 2D-to-3D conversion, our experiment does not describe the evaluation of time. But, there were times when the processing time was over 120 seconds in case that the resolution was high and camera motion was severe at an experimental
environment, i.e. in personal PC (Intel 2.8GHz CPU, 4GB Memory). As the processing time was added compared to the existing algorithm, it required improvement despite the need to accept the damage in terms of processing time. In all experiments, the size of block and the critical value of matching completion for the accuracy of block matching were set as $N=2, \varepsilon=100$.

5. Conclusion

2D-to-3D conversion is emerging as a way to solve the lack of 3D contents, which is blocking the development of 3D industry. We carried out a research on the key frame-based depth propagation, an automation technology propagating the depth of key frame to the other frames in 2D-to-3D conversion procedure, which was performed manually in most cases in the existing methods. The proposed method was the bilateral filtering, and unlike the existing proposed methods showing the errors due to the occlusion area caused by similar colors between object and background as well as motion, it is found that the proposed method can be used to predict the motion between frames more accurately using the reference depth and that the estimated depth map created as an actual result has a higher level of PSNR having the true depth than the existing methods.

Therefore, in future researches, we are planning to carry out a study on reflecting the method to increase the accuracy and the information to move object or camera toward the z-axis utilizing the other elements except color during the depth propagation process. At the same time, it seems to be necessary to make every effort to reduce the performing time more in order to process the video having higher resolution.

6. Acknowledgements

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program.

7. References

[10] Dichangsheng Wang, Ju Liu, Yannan Ren, Chuan Ge, Wei Liu, Yujun Li, "Depth propagation
based on depth consistency", In Proceeding(s) of International Conference on Wireless Communications & Signal Processing, pp.1-6, 2012.
[19] Ikuko Tsubaki, Atsutoshi Shimeno, Takeshi Tsukuba, Takeaki Suenaga, Masahiro Shioi, "2D to 3D conversion based on tracking both vanishing point and objects", In Proceeding(s) of IEEE Global Conference on Consumer Electronics, pp.110-114, 2012.