3-D VIDEO QUALITY IMPROVEMENT USING DEPTH TRANSITION DATA

Woo-Shik Kim *, Antonio Ortega

Signal and Image Processing Institute
Ming Hsieh Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

Jaejoon Lee, HoCheon Wey

Advanced Media Lab
Samsung Advanced Institute of Technology
Samsung Electronics Co., Ltd.

ABSTRACT

To improve rendered view quality in a 3-D video system, we propose to encode and transmit depth transition data, which represents, for each pixel in a frame, the location in between two existing views where the depth corresponding to that pixel changes. Given the highly localized and non-linear characteristics of rendered view distortion, it is possible to achieve better coding performance by providing this depth transition data only for subjectively important regions. In this paper, a method to apply the depth transition data to the view rendering procedure is proposed. Experimental results verify that improvements in subjective quality can be achieved by the proposed method.

Index Terms— 3-D video coding, multiview plus depth (MVD), view synthesis

1. INTRODUCTION

With recent advances in 3-D video system technology and 3-D content production, and the development of stereoscopic and auto-stereoscopic displays, compression for 3-D video applications is becoming increasingly important [1]. In this paper we focus on depth image based rendering (DIBR) [2, 3], where a depth image is encoded and transmitted, along with natural video captured in different view positions, in order to provide 3-D geometry information. Our previous research [4, 5] studies the relationship between the rendered view quality and the depth map distortion. As can be noticed in Fig. 1, the distortion in the rendered view caused by depth map distortion has non-linear characteristics (i.e., small depth errors can lead to very large intensity errors in the interpolated view) and is highly localized (i.e., more significant errors occur only in a small subset of pixels in a video frame).

To exploit these characteristics, in [6] we proposed the use of depth transition data (DTD), which represents, for each pixel in a frame, the location in between two existing views where the depth corresponding to that pixel changes. As will be shown here, DTD can be used to improve intermediate view rendering. Moreover, because of the localized nature of rendered view distortion, it is possible to achieve better coding performance by sending the DTD only for selected locations in the frame, instead of spending additional bits throughout the frame. While in our previous work DTD was used for post-processing in order to correct intermediate view rendering errors, in this paper we propose a new method to apply the DTD directly in the view synthesis procedure. Compared to [6], where DTD leads to improvements only when there are significant view interpolation artifacts, our new proposed scheme can be widely applied to various sequences. This is because DTD makes it possible to select the correct pixel in the reference view for view synthesis, leading to subjective quality improvements as will be shown experimentally.

The paper is organized as follows. In Section 2 the concept of the DTD is explained, and in Section 3 a new way to

Fig. 1. Temperature image of the absolute difference between the rendered view with and without depth map coding for the sequence Champagne Tower. Even though quantization is applied uniformly throughout the frame, its effect on view interpolation is much more significant near edges.
apply it to view rendering is proposed. Experiments are performed and the results are discussed in Section 4, followed by the conclusions in Section 5.

2. DEPTH TRANSITION DATA

As shown in Fig. 1, the rendered view distortion caused by depth map coding has non-linear, localized features. We focus on the case where one wishes to interpolate multiple intermediate views, for which no actual video is available [7]. In this case it would not be possible to transmit a residual signal to correct interpolation errors (since no “ground truth” video is available for intermediate view locations). Moreover, even if video is available for some intermediate camera position, the number of residues to be transmitted would increase linearly with the number of intermediate views, so that a residual based approach may not be efficient. Instead we propose to use the DTD, which can be used to improve interpolation at any intermediate camera position.

Consider a pixel location in two different views. Because of the different camera position the depth value at the same image coordinate can be different in each view. The DTD is the view position in between the two existing views at which the depth value for a given pixel location will change from that of the same pixel location in the right view to that of the same pixel location in the left view. Note that this implies that we do not need to send the DTD if a given pixel has the same depth in both left and right views. Fig. 2 shows an example, where a cube object is captured with three horizontally different camera positions as shown in Fig. 2(a). As the view index increases, the cube object moves to the left in the image frame. Therefore, for a given pixel location, we can trace how the depth value for that pixel changes as a function of the chosen intermediate camera position as shown in Fig. 2(b).

Note that conventional depth map information is provided for each reference view and is available for every pixel position, while DTD is associated to view pairs and is only transmitted for certain pixel locations. Since both standard depth information and DTD are transmitted, DTD allows us to spend additional bits in specific locations in order to improve view interpolation. Thus, we transmit DTD only for the subjectively important portions of the video, exploiting the highly localized nature of regions that are more sensitive to view interpolation errors (as shown in Fig. 1). The coding precision of DTD can also be easily adjusted depending on the desired density of intermediate views to be generated at the decoder (i.e., coarser quantization can be used if the number of interpolated views is small).

DTD can be obtained using the available depth maps at the encoder side by calculation of the depth transition position, \( t_x \), using the camera parameters and the distance, \( \Delta x_{\text{im}} \), from the given pixel location to where depth transition happens as follows:

\[
t_x = \frac{\Delta x_{\text{im}} + a_{x,p'} - a_{x,p}}{f_x} \cdot \frac{255}{a \cdot L_p (x_{\text{im}}, y_{\text{im}}) + 255b},
\]

where \( a = \frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} \) and \( b = \frac{1}{Z_{\text{far}}} \). Refer to Table 1 for the complete notations.

Then, \( t_x \) can be quantized to the desired precision according to the density of intermediate views to be synthesized. It is also possible to improve the precision of the DTD by use of video or depth map of intermediate view if available at the encoder side. For more details refer to [6].

3. APPLICATION TO VIEW RENDERING PROCESS

In our previous study [6], DTD is used to correct erosion artifacts that occur in the rendered view by using a post-processing after view rendering is completed. Since this is
done as a post-processing, DTD is not fully utilized to correct the distortion. For example, if the erosion is large, the inpainted region would not look natural. Also it cannot fix distortions other than the erosion along object boundaries. This can be improved by directly applying the DTD during the view synthesis process rather than correcting the distortion after the rendering process.

### 3.1. Selecting reference pixel using DTD

Fig. 3 illustrates the view synthesis process for a specific pixel location. When a reference pixel is warped into the target view, it is possible that a wrong pixel will be mapped to the target due to the depth map distortion (e.g., \( (x_{im}', y_{im}') \) is used for interpolation instead of \( (x_{im}, y_{im}) \) as in Fig. 3). The non-linear nature of the rendered view distortion can be understood from the example in Fig. 3: significant errors are produced if the chosen reference pixel belongs to a different object layer, e.g., a background pixel is used as reference for a pixel belonging to a foreground object. Thus similar depth errors may lead to very different errors in interpolation. DTD information allows us to signal explicitly that a specific pixel belongs to one of the two possible depth layers (corresponding to left and right images). Thus, in the example of Fig. 3 DTD will indicate that the intermediate pixel belongs to a foreground object.

Without knowing the true depth map value at the decoder side due to quantization, it is impossible to verify whether the reference is correct or not. However, by using the DTD we can verify whether it belongs to the same object layer as the target pixel.

![Fig. 3. Illustration of view synthesis process with possible depth error.](image)

As described in [6], an object layer map can be generated from the DTD for any intermediate view, indicating which object layer each pixel belongs to in the target view. The object layer can be formed according to the depth level of different objects. In the simulation, two object layers are used, which indicate foreground and background level. Note that this approach operates independently of view synthesis algorithms trying to generate depth layers in the reference views without providing additional information, since the proposed method generates the layer map in the target view by wisely spending additional bits to provide this to many intermediate views.

### 3.2. Detailed view synthesis procedure using DTD

A view synthesis is performed by warping reference pixels to the target view. When there is more than one reference pixel mapped to the same position in the target view, the synthesis can be done by a blending process. The blending process can be a weighted averaging using the distance between the reference and the target views as a weight. Or it is also possible to use the pixel which is nearer to the camera. If there is only one reference pixel mapped to the target view pixel position, its value can be directly copied to the target. If no reference pixel is mapped to a target pixel position, that location is regarded as a hole, and can be filled using an inpainting scheme. This subsection describes how DTD is applied for each step of view synthesis process. Fig. 4 provides the flowchart for this procedure.

When a pixel in the target view is synthesized, first the availability of the DTD for that pixel location is checked. If available, by using the DTD it is determined which object layer the pixel belongs to. Then, the pixels in the reference views are warped to the target view.

When a pixel is warped, we can know which object layer it belongs to in the reference view using its corresponding depth value. Therefore, it can be checked whether this warped pixel from the reference view (reference pixel) belongs to the same object layer as that of the target view pixel. If the reference pixels belong to the same object layer as the target, they are regarded as valid and used for the view synthesis. Conversely, if the reference pixel does not belong to the same object layer as the target pixel, this implies it is not safe to use this reference pixel for the view synthesis.

Then, there can be some pixel position in the target view, where no reference pixel is mapped from the reference views. In this case, it is allowed to use the reference pixels in different object layer only when its corresponding depth value is close to that of the other reference view. This is the case where the transition has happened but with a small amount of depth change, which can be set as a threshold value to indicate the tolerance range with a unit of depth map value. In our simulation, a fixed threshold value is applied for all the sequences to simplify the simulation; however, it would be possible to adapt the threshold value according to global and local sequence characteristics such as camera parameters, contrast in local area, etc., to improve the performance. Note that in this case the DTD does not need to be provided, therefore, we do not need to code and transmit it, but the encoder and decoder
The performance of the DTD is related to various parameters such as camera settings and scene characteristics. For example, if there is a large camera distance between views, and the object in the scene is near, the amount of geometry error caused by depth map distortion will be large, since the pixel displacement during the warping procedure will be large. The warping distance can be written as:

$$d_{x,im} = f_x \cdot t_x / z,$$

where $d_{x,im}$ is translation in the image coordinate corresponding to camera distance $t_x$, $f_x$ is focal length divided by the effective pixel size, and $z$ is depth value of the object in the scene. When $d_{x,im}$ is large, there will be more chances for the wrong reference pixel to belong to different foreground/background region than the target. In this case significant improvement can be achieved by using the DTD by eliminating these wrong reference pixels.

The contrast across the boundary between the foreground and the background also affects the performance. When a wrong reference pixel with high contrast is used, this will cause noticeable distortion. In this case, there will be more room to improve the performance by correcting the distortion using the DTD. The size of the foreground object can affect the performance as large size will provide more area to improve the performance. The performance of the DTD is also related to the bitrate of the depth map. If the depth map is coded in very high bitrate with very high quality, the portion of the DTD bitrate in the total bitrate would be small, but there could be not so much chance for the DTD to reduce the distortion. On the other hand, if the depth map bitrate is very low, the portion of the DTD bitrate will increase, but there will be large room to improve the quality. Therefore, it is desirable to use the DTD according to these parameters to achieve the best performance.

### 4. EXPERIMENTAL RESULTS

The proposed method is evaluated with experiments using various test sequences. Table 2 lists the test sequences used in the experiments with the view indices of left and right reference views and the target view. $d_{x,im}$ in (2) is also calculated with setting $z = Z_{\text{near}}$ for each sequence, and included in Table 2 to compare different camera settings among various sequences.

#### Table 2. View index of test sequences.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Left</th>
<th>Target</th>
<th>Right</th>
<th>$d_{x,im}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cafe</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>180.0</td>
</tr>
<tr>
<td>Mobile</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>43.5</td>
</tr>
<tr>
<td>Champagne Tower</td>
<td>37</td>
<td>39</td>
<td>41</td>
<td>146.1</td>
</tr>
<tr>
<td>Balloons</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>50.0</td>
</tr>
<tr>
<td>Newspaper</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>100.0</td>
</tr>
<tr>
<td>Pantomime</td>
<td>37</td>
<td>39</td>
<td>41</td>
<td>76.2</td>
</tr>
<tr>
<td>Lovebird1</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>67.1</td>
</tr>
<tr>
<td>Kendo</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>50.0</td>
</tr>
</tbody>
</table>

The DTD is calculated using the left and right reference views first by generating their object layer maps with two levels (foreground and background) followed by calculating the transition position using (1) for the region where transition happened between the reference views. The foreground/background determination is performed based on the depth map value using k-means clustering. The fixed threshold value of the tolerance range is set to 30 for all the sequences. To cover the area where transition happens more
than once between the reference views, the target view depth map is used, and the first transition position is recorded. Then, it is quantized to represent the mid-position between the reference views, and losslessly compressed using H.264 with CABAC encoding.

The proposed method to apply the DTD to view synthesis is implemented using the MPEG view synthesis reference software (VSRS) 3.0 [8]. Fig. 5 shows the rate-distortion curves with and without the DTD, where both depth map and video are compressed using H.264 with QP values of 24, 28, 32, and 36. The inter-view prediction is used to code both video and depth maps to reduce the bitrate. Two reference views are coded and the middle point view between them is synthesized. The PSNR of the rendered view is calculated compared to the ground truth. The x-axis is the bitrate to code reference video and depth maps. In case of “Depth transition data”, the bitrate to code the DTD is added. The y-axis is the PSNR of Y component.

In Fig. 5, it can be observed maximum PSNR improvement of 2 dB in the Champagne Tower, and 1.5, 0.6, and 0.5 dB in the Mobile, Cafe, and Balloons, respectively. Also there are smaller PSNR improvements in the Newspaper and Pantomime sequences. No gain is observed in the Lovebird1 and Kendo sequences. For the Cafe and Champagne Tower, large value of $d_{x,im}$ contributes to the improvement, and large size of foreground objects in the Champagne Tower increases the performance improvement. For the Mobile sequence, high contrast across the object boundary contributes to the improvement. In case of the Mobile sequence, foreground object is small and the contrast is not strong, so no improvement is observed. For the Kendo sequence, the depth map is noisy, so it does not provide clear object boundary. In this case, it would be possible to improve the performance by generating the DTD using other information such as reference video. We leave this as a future study.

In all the sequences, the bitrate is increased for the DTD case due to the bitrate to code the DTD. Table 3 shows the relative percentage of the total bitrate used for DTD. Since it is coded losslessly, the percentage of rate used for DTD increases as QP increases.

Fig. 6 show the subjective quality comparison between the rendered view without (upper row) and with (bottom row) the DTD. In Cafe, it can be noticed the wrong pixel mapping in man’s hair area is corrected using the proposed method. Again, in Mobile and Champagne Tower, wrong reference pixel mapping in the foreground object is corrected, and clear object boundary can be observed.

5. CONCLUSIONS AND FUTURE WORK

The depth transition data is proposed as a new 3-D video data format to achieve better coding efficiency with improved rendered view quality by exploiting localized and non-linear characteristics of the rendered view distortion. The experimental results show that the subjective quality improvement can be achieved by applying the depth transition data to the view synthesis process.

In the future, we will further analyze various factors affecting the rendered view quality, so that optimal way can be found to compensate the distortion using the depth transition data. In addition, we will investigate how to improve the performance when a depth map is noisy.

6. REFERENCES


Table 3. Bitrate occupancy of video, depth map, and DTD in total bitrate for two reference views.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>QP</th>
<th>Bitrate distribution (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Video</td>
</tr>
<tr>
<td>Cafe</td>
<td>24</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>49</td>
</tr>
<tr>
<td>Mobile</td>
<td>24</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>72</td>
</tr>
<tr>
<td>Champagne Tower</td>
<td>24</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>63</td>
</tr>
</tbody>
</table>
Fig. 5. Rate-distortion curves with and without depth transition data.

Fig. 6. Subjective quality comparison: conventional method (upper row) and the proposed method with depth transition data (bottom row); Cafe, Mobile, and Champagne Tower from left to right (QP 24).