3-D VIDEO CODING 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
Multimedia Lab
Samsung Advanced Institute of Technology
Samsung Electronics Co., Ltd.

ABSTRACT
The objective of this work is to develop a new 3-D video coding system which can provide better coding efficiency with improved subjective quality as compared to existing 3-D video systems. We have analyzed the distortions that occur in rendered views generated using depth image based rendering (DIBR) and classified them in order to evaluate their impact on subjective quality. As a result, we found that depth map coding distortion leads to “erosion artifacts” at object boundaries, which lead to significant degradation in perceptual quality. To solve this problem, we propose a solution in which depth transition data is encoded and transmitted to the decoder. Depth transition data for a given pixel indicates the camera position for which this pixel’s depth will change. A main reason to consider transmitting explicitly this information is that it can be used to improve view interpolation at many different intermediate camera positions. Simulation results show that the subjective quality can be significantly improved by reducing the effect of erosion artifacts, using our proposed depth transition data. Maximum PSNR gains of about 0.5 dB can also be observed.

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

1. INTRODUCTION
High quality auto-stereoscopic displays create new 3-D multimedia experiences for the users. However, they require transmission/storage of multiple video sequences representing a scene captured from different view positions. Thus the need for efficient compression tools for this kind of content is clear [1]. For this purpose multiview video plus depth formats are being developed, where only selected views are coded, and other views are interpolated at the decoder using depth image based rendering (DIBR) techniques [2]. While DIBR-based approaches are advantageous in terms of overall bitrate (since not all views need to be transmitted), they also pose new challenges, since the quality of decoded views depends on the interpolation process.

In this paper we start by focusing on the different types of distortion affecting the DIBR process. Clearly, the quality of decoded frames that are used for view interpolation is an important factor, but since they are likely to be coded using standard tools (e.g., H.264/AVC) it is relatively simple to control their distortion. Instead, we focus on how the quality of depth maps transmitted to the decoder affects overall quality. Previous work has studied improved coding tools for depth information, especially with a goal of preserving edge information [3, 4, 5], as well modifications to distortion metrics used for depth map coding that take explicitly into account the impact depth map representation has on the interpolated [6, 7]. The key novelty in this paper is to propose a new 3-D video format in which additional information is sent to the decoder (in addition to video from selected views and depth information). We choose to transmit to the decoder information about depth transition data. To provide some intuition this type of data consider a scenario where reference views are transmitted and depth information for them is available. Focusing on a specific pixel in one view, given its corresponding depth information, we can determine its corresponding location in the other view. If depth information for this second view indicates a significantly different depth value than for the first view, this is a sign that an occlusion has occurred. For these situations we transmit to the decoder the location at which this transition occurred (from an object at a certain depth to another at a different depth). Providing explicit information about where this transition occurs has several advantages. First, it leads to improved interpolation for arbitrary intermediate camera positions (we will know on which side of the occlusion location the intermediate view is). Second, in cases where no depth information is available for intermediate views, depth transition data could also be derived from depth information at the transmitted views. However, by explicitly representing transition information we are able to improve quality (by using more bits) at those locations where this information matters most (where occlusions occur). Finally, in cases where depth information is available for intermediate views (but is not going to be transmitted), the depth transition data can be corrected, so that improved interpolation can be achieved without having to transmit additional depth maps.

Our preliminary results indicate that our proposed depth transition data representation can be exploited to achieve significant perceptual gains in the rendered frames, with average PSNR gains of up to 0.5 dB. The rest of the paper is organized as follows. In Section 2 different types of distortion affecting the rendered view are analyzed. In Section 3 we propose new 3-D video format using depth transition data. Experiments are performed and the results are discussed in Section 4, followed by the conclusions in Section 5.

2. ANALYSIS OF DISTORTION IN RENDERED VIEW
There are multiple sources of distortion introduced in the DIBR process. First, distortion in the reference views can directly affect rendering quality, since these reference frames are used for interpolation. Second, degradation in the rendered view may be due to the synthesis procedure itself. For example, there may be occlusions, or differences in illumination between views used for interpolation. In
these cases, there will be errors even if the depth map is encoded losslessly.

Fig. 1. Subjective quality comparison of the rendered view using the Ballet sequence. (a) rendered view 5 using views 4 and 6 as references, using the original depth map, without distortion due to depth map coding, and (b) same view rendered with a decoded depth map.

In this paper we focus on distortion due to depth map inaccuracy, i.e., resulting from coding distortion introduced by a lossy representation. As discussed in our previous work [6, 7], distortion in depth values leads to geometry errors in performing the interpolation. To illustrate the perceptual impact of these errors, refer to Fig. 1, where interpolation based on the original depth map and a decoded one are shown. In Fig. 1(a) we start by noting that the view synthesis process causes errors even without any coding involved. For example, high frequency components tend to be blurred throughout the image due to the blending process during the view synthesis. There are also very clear errors in the occluded regions, for which the blending is not used, and one reference video is used instead to fill those regions. This is due to differences in illumination between different views (if no averaging is performed across views, the illumination of the interpolated view will be similar to that of one of the reference view, which may not reflect the true illumination). Finally, there exists distortion around object boundaries. This is due to occlusion and/or depth map estimation error.

Fig. 1(b) illustrates the effect of depth map encoding. Note that severe distortion occurs along object boundaries, where the foreground object boundary is “eroded” by the background area, leading to an erosion artifact. In areas with sharp changes in depth (e.g., at the boundary between a foreground object and background), compression is likely to produce artifacts (e.g., ringing) in the decoded signal. The original boundaries of the object may be distorted, leading to both erosion and dilation of the area corresponding to the object in the depth map. In an eroded region pixels originally corresponding to the object will now have depth values associated to the background. Correspondingly, in the dilated region parts of the background pixel data will be associated to foreground depth values. When rendering a view with this distorted depth map, the regions corresponding to foreground depth (i.e., the object minus the eroded regions) will be warped with larger displacement than the background region. Since the eroded region no longer has foreground depth value it will not move with the foreground object, therefore the object that is displaced will itself be eroded in its boundary. Instead, note that the error is less visible for dilated regions. This is because in those dilated regions, background intensity data is associated to foreground depth and is displaced along with the object. This will not cause very visible errors in the object shape, especially in cases, such as Fig. 1, where the background is uniform. These artifacts due to dilation will be more visible when background is not uniform, as some of the area surrounding the object will correspond to background in the original view.

In what follows, we present our proposed depth transition data and focus on how it can be used to remove erosion artifacts, since these tend to be more visible. While we focus on this particular class of distortion at this point, we plan to make use of the same data format to address other artifacts as well.

3. NEW 3-D VIDEO FORMAT

One possible approach to remove erosion (and other) interpolation artifacts would be to provide additional information for each intermediate rendered view. A simple example of this would be to synthesize views at the encoder and transmit a residue between the synthesized view and the original captured video. This solution is not attractive because the required overhead will increase with the desired number of possible interpolated views. Instead, our goal is to provide auxiliary data that complements depth information and can be used to improve rendering of multiple intermediate views. With this approach the same auxiliary data can be sent, giving users maximum flexibility in determining which view should be interpolated.

We propose transmitting depth transition data for specific pixel locations, as illustrated by Fig. 2. A cube object is captured with 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 changes for that pixel as a function of the chosen intermediate camera position as shown in Fig. 2(b).

Fig. 2. Depth transition example (a) cube object captured in horizontally different camera positions (b) depth value transition curve at a specific image coordinates.

Compared to the conventional depth map information which is provided for every reference view separately, the advantage of the proposed depth transition data is that once it is generated, a single data set is needed to aid rendering any arbitrary view position. In addition, we can provide depth transition data only (or with more precision) for subjectively important portions of the video. It is also simple to control encoding precision for depth transition data, as a function of the minimum spatial spacing between intermediate cameras. Thus, a coarse representation can be used if only a few intermediate views need to be interpolated.
where depth transition happens. Assume a horizontal parallel camera arrangement, which implies that $R = I$ (identity matrix). To calculate $A_p R_p^{-1}$, we define the intrinsic matrix $A$ as

$$
A = \begin{pmatrix}
    f_x & 0 & 0 \\
    0 & f_y & 0 \\
    0 & 0 & 1
\end{pmatrix},
$$

where $f_x$ and $f_y$ are the focal length divided by the effective pixel size in horizontal and vertical direction, respectively, and $(o_x, o_y)$ is the coordinate in pixel of the image center (the principal point). If we assume the same focal length for both cameras at $p$-th and $p'$-th views, (1) will become as (2).

$$
\begin{align*}
\begin{pmatrix}
    x'_p \\
    y'_p \\
    1
\end{pmatrix}
&= A_p R_p^{-1} \begin{pmatrix}
    x_p \\
    y_p \\
    1
\end{pmatrix} + \frac{1}{Z_p(x_p, y_p)} A_p R_p \{ T_p - T_p' \} \\
\begin{pmatrix}
    x'_p \\
    y'_p \\
    1
\end{pmatrix}
&= \begin{pmatrix}
    x_p + o_x, p - o_x, p' \\
    y_p + o_y, p - o_y, p' \\
    1
\end{pmatrix} + \frac{1}{Z_p(x_p, y_p)} A_p \{ T_p - T_p' \}
\end{align*}
$$

$$
\Delta x_p = x'_p - x_p = o_x, p - o_x, p' + \frac{L_p(x_p, y_p)}{255}, \frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} + \frac{1}{Z_{\text{lat}}} \cdot f_x \cdot t_z
$$

3.1. Generation of depth transition data

Depth transition data is generated for each pixel location by tracing its depth map value change. For simplicity we transmit (with appropriate spatial resolution) the location at which a given pixel alters from foreground to background or vice versa. Depth values are still sent for the reference frames, so that values corresponding to foreground and background can also be determined.

First, it is necessary to set a criterion for foreground/background determination for the reference views. There have been various research on foreground/background separation, e.g. using video motion information, segmentation based method, etc. In this work, as a preliminary approach the depth map is used for the separation. For example, a middle point of the depth map value range, i.e. the average of maximum and minimum depth map values can be used as a threshold value, and for each pixel location, if its depth map value is smaller than the threshold, it will belong to the background, and vice versa. The threshold value can be adjusted for better separation of the foreground and background. Note that more advanced algorithm will help to improve the performance of our proposed method.

In the simple case when depth maps for the intermediate views are available at the encoder, but not at the decoder, the depth transition data can be generated directly by generating the foreground/background binary map from the available depth map. Depth transition data simply records for each pixel the intermediate view index at which transition occurs.

Next, consider the case where no depth/video information is available for intermediate views. The camera position where the depth transition happens can be estimated using the camera parameters. As shown in our previous work [6, 7], given per pixel depth values, we can map a pixel position into a point in the world coordinates, and that position can be remapped in another camera coordinates that belongs to the view to be rendered. Specifically, if we map a point in the $p$-th view of which the camera parameters are $A_p, R_p,$ and $T_p,$ to the $p'$-th view with parameters of $A_{p'}, R_{p'},$ and $T_{p'},,$ the image coordinates in the $p'$-th view can be represented as in (1), where $A$ is the intrinsic camera matrix, $R$ is the rotation matrix, and $T$ is the translation vector, and $Z$ is a depth value.

Now, based on our previous derivation of point mapping, we show how it is possible to calculate the camera position at which depth transition happens. Assume a horizontal parallel camera arrangement, which implies that $R = I$ (identity matrix). To calculate $A_p R_p^{-1}$, we define the intrinsic matrix $A$ as

$$
A = \begin{pmatrix}
    f_x & 0 & o_x \\
    0 & f_y & o_y \\
    0 & 0 & 1
\end{pmatrix},
$$

where $f_x$ and $f_y$ are the focal length divided by the effective pixel size in horizontal and vertical direction, respectively, and $(o_x, o_y)$ is the coordinates in pixel of the image center (the principal point). If we assume the same focal length for both cameras at $p$-th and $p'$-th views, (1) will become as (2).

Under the parallel camera setting assumption, there will be no disparity change other than in horizontal or $x$-direction. Therefore, the disparity $\Delta x_p$ can be expressed as

$$
\Delta x_p = x'_p - x_p = o_x, p - o_x, p' + \frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}} \cdot f_x \cdot t_z,
$$

where $t_z$ indicates the camera distance in horizontal direction. The relationship between the actual depth value and 8-bit depth map value is

$$
L(x, y) = \left( \frac{x}{Z_{\text{near}}} - \frac{x}{Z_{\text{far}}} \right) \times 255,
$$

where $Z_{\text{near}}$ and $Z_{\text{far}}$ are the nearest and the farthest depth value in the scene, which would correspond to value 255 and 0 in the depth map $L$, respectively. By plugging this into (5), we can get (3).

Therefore, if we know the camera distance, $t_z$, we can calculate the disparity, $\Delta x_p$, and vice-versa. Therefore, if we use the disparity as the horizontal distance from the given pixel location to where depth transition happens, we can find the exact view position where the transition happens. The horizontal distance can be found by counting the number of pixels from the given pixel to the first pixel for which the depth map value difference with respect to the original pixel exceeds a preset threshold value. Then using this distance as the disparity, $\Delta x_p$, we can estimate the view position at which depth transition occurs as:

$$
t_x = \frac{\Delta x_p - o_x, p - o_x, p'}{f_x} \times 255,
$$

where $a = \frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}$ and $b = \frac{1}{Z_{\text{lat}}}$. Then, $t_x$ can be quantized to the desired precision and transmitted as auxiliary data.

3.2. Correcting erosion artifact

This section describes the procedure to correct the erosion artifact using the proposed depth transition data. To render a view at an arbitrary view point, we can use the depth transition data to determine whether each pixel belongs to the foreground or background. For example, at a specific pixel location, if it belongs to the foreground in the left reference and to the background in the right reference, we know that the depth transition happened between these two views. If the depth transition data indicates that this transition happens before the target view to be rendered, the pixel location would belong to the background in the target view. If the depth transition data indicates the transition happens after the target view, the pixel location
would belong to the foreground in the target view. When the transition happens in the other way, i.e., from background to foreground, this decision will be reversed. We assume that the distance between cameras is small enough so that there is at most one depth transition between two reference views.

Once the foreground/background map for the target view is generated using the depth transition data, the erosion correction can be performed for a given local area, e.g., 8 × 8 block in the rendered view with erosion artifact. Each area can overlap and/or have adaptive size for better performance. First, for the given block a background average is calculated using the pixels belonging to the background. Then, each foreground pixel is compared to the background average. If a pixel is close to the background average, it is classified as an outlier or eroded pixel. Then, foreground average is calculated using the pixels without outliers. Finally, the eroded pixel values are replaced with the foreground average.

To replace the eroded pixel value, it would be also possible to use the nearest foreground pixel value which is not the outlier, or utilize the pixel values in the reference video. In addition there can be other ways of using the depth transition data to improve the DIBR process. We will further investigate these possibilities as a future work.

4. EXPERIMENTAL RESULTS

To evaluate the proposed depth transition data and its application to the erosion artifact correction, experiments are performed using Ballet and Breakdancers test sequences [2] and the View Synthesis Reference Software (VSRS) 3.0 [8]. Fig. 3 shows the rate-distortion curve comparison where the 5th view is synthesized using 4th and 6th views as references. The reference video and depth maps are coded using H.264 with same QP values of 24, 28, 32, and 36 to generate the curve, and Y PSNR is calculated by comparing the synthesized view to the original 5th view. For ‘synthesized view with AUX’, the same synthesis is performed followed by the erosion correction using the depth transition data. For the erosion artifact correction, block size of 8 × 8 is used without overlapping. The bitrate to code the depth transition data is approximately estimated by coding the foreground/background binary map of the target view, and added to the bitrate to code two sets of reference video and depth map. It may be possible to reduce the bitrate required to encode the depth transition data by exploiting the reference depth map information. This will be further investigated and implemented as a future work. From the curves, we can notice that there are moderate bitrate increases due to the depth transition data, while there are noticeable PSNR gains for the Ballet sequence, and smaller gains for the Breakdancers sequence. The reason for the smaller gain in the Breakdancers sequence is that some objects in the scene are close to the background in depth, so that not many portions of the scene belong to the foreground. In addition, the difference between the foreground and background pixel value is not as significant as the Ballet sequence, which reduces the PSNR gain.

Fig. 4 shows the subjective quality improvement with the proposed method. In Fig. 1 (b) the erosion artifact is clearly visible which degrades the subjective quality greatly, and in Fig. 4 it can be noticed that the eroded area is fixed using the proposed method, thus noticeable subjective quality improvement is achieved.

5. CONCLUSION AND FUTURE WORK

We have developed the new 3-D video format by providing the depth transition data in addition to the existing video plus depth data. The depth transition data indicates where the background/foreground transition happens, therefore is applicable to correct the depth map distortion. While the conventional DIBR based scheme needs to provide the depth map information to every reference views, the advantage of the proposed depth transition data is that one such data set can be applied to interpolate multiple views at arbitrary positions. By using the proposed depth transition data, erosion artifacts due to depth map distortion can be greatly reduced, thus improving the subjective quality of the rendered view significantly.

As we continue this work, we will further investigate how to efficiently represent and encode the depth transition data. The erosion correction algorithm using the depth transition data will be further studied to achieve better subjective quality improvement.

6. REFERENCES