DEPTH IMAGE-BASED RENDERING WITH SPATIO-TEMPORALLY CONSISTENT
TEXTURE SYNTHESIS FOR 3-D VIDEO WITH GLOBAL MOTION

Martin Köppel†, Xi Wang*, Dimitar Doshkov*, Thomas Wiegand†‡, and Patrick Ndjiki-Nya†

†Image Communications Group, Technical University of Berlin, Einsteinufer 17, 10587 Berlin, Germany

‡Technical University of Berlin, Einsteinufer 37, 10587 Berlin, Germany

ABSTRACT

Depth image-based rendering (DIBR) techniques are advanced tools in 3-D video (3DV) applications that are used to synthesize a number of additional views in a multiview-video-plus-depth (MVD) representation. The MVD format consists of video and depth sequences for a limited number of original camera views of the same scene. An inherent problem of the view synthesis concept is given by image information that is occluded in the original views and becomes visible, in the extrapolated views. To handle these disocclusions, we propose a DIBR algorithm with advanced inpainting methods. Our renderer enhances visual experience by taking spatial and temporal texture consistency problems into account. In order to compensate the global motion in a sequence, image registration is incorporated into the framework. The proposed method shows objective and subjective gains compared to the state-of-the-art.

Index Terms— Inpainting, depth image-based rendering (DIBR), 3-D video, texture synthesis

1. INTRODUCTION

In recent years, 3-D video (3DV) technology has reached the wide market. The 3-D Blue-Ray and stereo displays entered the shops and the living rooms. Despite the high standard that 3DV has reached today, there is still room for improvements. Moreover, new technologies such as autostereoscopic multiview displays require further research. The common representation format used for autostereoscopic displays is multiview-video-plus-depth (MVD). The MVD format consists of video and depth sequences for a limited number of original camera views of the same scene. In order to generate additional virtual views depth image-based rendering (DIBR) techniques are used. A significant problem in DIBR techniques is that texture areas become disoccluded (holes) in the virtual views, especially in the extrapolated views beyond the viewing range of the original cameras. Three general methods have been proposed in the literature to handle such holes. First, the depth maps are preprocessed in a way that no disocclusions occur. Usually, the depth map is smoothed, using a symmetric [1] or an asymmetric filter [2], to lower gradients. This method gives good results, when small baselines have to be compensated. Nevertheless, foreground and background textures are distorted, especially in foreground-background transitions, during the warping procedure. The second way to close disocclusions is to cover them with plausible, known image information. Suitable filling techniques such as line-wise filling [3], inpainting methods [4, 5], bilateral filtering [6], and texture synthesis [7] are often used.

Alternatively, image domain warping can be utilized to compute the virtual views. Applying the latter method, holes are covered by distorting non-salient image regions [8].

Another problem in DIBR is to maintain temporal consistency in the covered texture areas. First approaches that handle this problem have been published in [7], [9] and [10]. In [7] and [9] a mosaic/sprite is used to store background information from temporally neighboring frames for further reuse during the filling process. Nevertheless, these approaches are restricted to sequences with static backgrounds. Chen et al. [10] assume that the original views are encoded with H.264/AVC and use the motion vectors from the bit stream to find appropriate information in temporally shifted frames. However, the motion vectors in H.264/AVC are sparse and encoder optimized. Therefore, the motion vectors do not necessarily correspond to the real motion. Hence, only little PSNR and SSIM gains are obtained in [10]. Due to the rising necessity of extrapolated views in 3DV, the Moving Pictures Experts Group (MPEG) established first experiments to explore the extrapolation capabilities of DIBR algorithms [11].

In this paper a new approach to tackle disocclusions in virtual views is presented. Temporal consistency is achieved by using image information from previous and subsequent frames. By utilizing a robust image registration tool [12], global background motions for a set of neighboring frames can be compensated. Spatial consistency in virtual views is ensured by a specific warping step. This approach is developed for the MVD representation but can be used as well for the video-plus-depth (V+D) format.

2. PROPOSED FRAMEWORK

The proposed framework for DIBR with spatially and temporally consistent texture synthesis is shown in Fig. 1. In a first step, the original views and the associated depth maps are warped to the outermost (application dependent) left and/or right position beyond the original camera range by using the algorithm proposed in [5]. Then, holes in the depth maps are filled and a set of previous and subsequent frames is registered to the position of the actual frame by using the image registration method proposed in [12]. Only background image information is utilized, to compute the required transformation parameters. Next, the holes in the current frame are updated with reliable image information from the registered previous and subsequent images. The remaining holes in the frame are initialized and subsequently filled using texture synthesis [7]. In a final warping step, the remaining virtual views are interpolated between an outermost already filled virtual camera and an original camera or between two original cameras by using the method proposed in [5].

In the following, the image to be filled will be denoted as $F_n$, the
associated depth map as $D_n$. Holes in an image and a depth map will be referred to as $\Omega_F$ and $\Omega_D$ respectively. Registered frames and associated depth maps will be denoted as $F_{n\pm i}$ and $D_{n\pm i}$, $i = [1, f]$ with $i, f \in \mathbb{N}$ (cf. Fig. 2). The original texture in a frame will be $F^o \subset F \Omega_F$ and the synthesized texture in a frame will be denoted as $F^r$. A pixel position in an image or in a depth map will be denoted as $p$.

3. DEPTH MAP FILLING

The depth maps used in this framework are represented as gray-scale images, quantized to 8-bit. They give the geometric placement of objects in a scene. This characteristic is used for the image registration step, the frame update procedure and texture synthesis. The method proposed in [7] is used to fill the uncovered areas $\Omega_D$ in the depth maps. During the filling process a k-means clustering procedure detects background- and foreground areas and a threshold ($c_{min}$) is automatically calculated. Objects whose depth values are larger than $c_{min}$ are considered as foreground (for further details please refer to [7]).

4. IMAGE REGISTRATION

Image registration is a process of transforming different sets of data onto one common coordinate system by geometrical mapping. In order to cope with the global motion in our application scenarios, an area-based image registration approach [12] is incorporated into the overall framework.

As shown in Fig. 2, a number $f$ of previous and subsequent frames and the associated depth maps are registered to the current frame $F_n$. Due to the fact, that only the global motion in the background areas should be compensated, foreground objects must be excluded from the registration step. Based on the depth values in the depth map and $c_{min}$, foreground areas/objects can be detected. Hence, only background image information is used to compute the affine transformation matrices. To decide, whether the image registration was successful, the PSNR between $F_n$ and the registered frame is determined for the background region used in the registration step. If the PSNR, measured in the luminance channel, lies above a chosen threshold ($t_{psnr}$), the registered frame is considered for frame update.

Algorithm 1: Pseudo code for image update

```plaintext
for i ← 1 : f do
  for each pixel position $p$ in the $\Omega_F$ do
    if $F_{n-i}(p)$ and $F_{n+i}(p)$ are in depth range (cf. Eq. 1) then
      Compute new texture value (cf. Eq. 2);
    else if $F_{n-i}(p)$ or $F_{n+i}(p)$ is in depth range (cf. Eq. 1) then
      Choose texture value in depth range;
    else
      Take no action;
  end for
end for

for i ← 1 : f do
  for each remaining pixel position $p$ in the $\Omega_F$ do
    if $F_n(p)$ is in depth range (cf. Eq. 1) then
      Choose synthesized texture value;
    else
      Take no action;
  end for
end for
```

In a set of subjective evaluations, it was found that at least $t_{reg}$ (80%) percent of background information needs to be present to obtain stable and useful registration results. Thus, $c_{min}$ is iteratively refined until $t_{reg}$ percent of the background information can be used.

5. FRAME UPDATE AND TEXTURE FILLING

Due to the fact that the depth map represents the geometric placement of the objects relatively to the camera position, image information with an appropriate disparity can be selected. Thus, the synthesized depth values in $\Omega_D$ (cf. Sec. 3) are utilized, to decide pixel-wise, whether the image information from temporally surrounding registered frames can be used to fill the corresponding hole $\Omega_F$ in the texture frame. The update procedure is shown as pseudo code in Algorithm 1. Image information from temporally closer frames is regarded first (cf. Fig. 2, first solid lines then dashed lines). Hence, a frame pair (one previous and one subsequent) is considered simultaneously for updating (cf. Fig. 2). Original texture is used primarily and synthesized data as fall-back in the update process. An available pixel position $F_{n\pm i}(p)$ in a registered frame is considered for updating $F_n(p), p \in \Omega_F$ if the associated registered depth value $D_{n\pm i}(p)$ is in the required depth range (Algorithm 1). This can be formalized as follows:

$$D_n(p) - t_{dr} < D_{n\pm i}(p) < D_n(p) + t_{dr}, \quad (1)$$

where $t_{dr}$ is a system parameter, to account for small depth changes between temporally adjacent frames. If two values $F_{n-i}(p)$ and $F_{n+i}(p)$ are considered to update the missing pixel position, $F_n(p)$ is computed as follows (cf. Algorithm 1):

$$F_n(p) = \frac{F_{n-i}(p) + F_{n+i}(p)}{2}, \quad p \in \Omega_F \quad (2)$$

In order, to fill the remaining disocclusions in $F_n$, the method proposed in [7] is used. First, small holes up to a certain threshold (50 pixel are used as suggested in [7]) are closed by using advanced cloning methods. Bigger holes are first initialized and then optimized with patch-based texture synthesis. To account for intensity variation between adjacent patches covariant cloning is incorporated (for further details please refer to [7]).
Table 1: PSNR results by the proposed framework, VSRS and VSRS-Alpha-Gist.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Cam.</th>
<th>Prop.</th>
<th>VSRS</th>
<th>Gist</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>5 → 1</td>
<td>42.0708</td>
<td>37.9720</td>
<td>34.0833</td>
</tr>
<tr>
<td>S1</td>
<td>5 → 9</td>
<td>43.5635</td>
<td>38.3514</td>
<td>37.5000</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 1</td>
<td>34.8230</td>
<td>33.8626</td>
<td>33.0428</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 9</td>
<td>32.2160</td>
<td>27.8509</td>
<td>31.1527</td>
</tr>
<tr>
<td>S3</td>
<td>6 → 7</td>
<td>35.5210</td>
<td>35.3976</td>
<td>33.0428</td>
</tr>
<tr>
<td>S4</td>
<td>3 → 1</td>
<td>41.5517</td>
<td>40.5210</td>
<td>38.0918</td>
</tr>
<tr>
<td>S4</td>
<td>5 → 7</td>
<td>28.9286</td>
<td>28.8419</td>
<td>28.1672</td>
</tr>
</tbody>
</table>

6. FINAL WARPING

Fig. 3 illustrates the final as well as the initial warping. The black cameras are the original views and the gray cameras are the outermost synthesized virtual views. All necessary but still missing virtual views (cf. Fig. 3, white cameras) are interpolated between the original and outermost cameras by using the algorithm described in [5]. The number of virtual views can be adjusted depending on the specific device or application.

With the final warping procedure, spatial continuity and stability is achieved, especially in the synthesized areas $F^o$. The overall complexity decreases due to the fact that only the outermost views need to be rendered in a complex way in order to get visually pleasing results. The complexity of the proposed method depends on the size of $\Omega_F$. The larger $\Omega_F$ is, the more processing operations need to be done.

7. EXPERIMENTAL RESULTS

The proposed approach is compared to the MPEG view synthesis reference software (VSRS, version 3.5) [5] and to VSRS-Alpha-Gist [6]. For the evaluation of the proposed algorithm, four MPEG MVD test sequences are used: “GT-Fly” (S1, 249 frames), “Undo dancer” (S2, 249 frames), “Poznan Hall2” (S3, 200 frames) and “Balloons” (S4, 300 frames). S1, S2 and S3 have a resolution of 1920 × 1088 samples, while S4 has a resolution of 1024 × 768 samples. We evaluated our framework with the following parameter setting: $f = 2$ frames, $t_{post} = 21$ dB, $t_{reg} = 80\%$ and $t_{obs} = 15$. For the texture synthesis method, the parameter setting proposed in [7] is utilized. Only the patch size is changed to 21 due to the fact that we use sequences with higher frame resolutions. We evaluate the extrapolation capabilities of our proposed approach as follows: the outermost virtual view is rendered from the utilized original sequence and its associated depth map, as shown in the camera column (Cam.) in Table 1 (i.e. “5 → 1” means: camera 1 rendered from original camera 5). Then the outermost camera and the original camera, are used together to interpolate at least one virtual view in between. We compare the results with VSRS and VSRS-Alpha-Gist, which are rendered only from the original camera. The outermost views are evaluated by measuring the PSNR in the luminance channel between the rendered frames and the original data that are available in that case. The interpolated results between an original view and a virtual view are evaluated subjectively (cf. Fig. 4).

The objective results given in Table 1 correspond to the mean PSNR over all pictures for the outermost rendered sequence. The best result for every sequence is highlighted through bold face type. For S1, S2 and S3 the proposed approach performs better than VSRS and VSRS-Alpha-Gist for all considered sequences (cf. Table 1). For S4, “3 → 1”, the proposed approach performs better than VSRS-Alpha-Gist but worse than VSRS. The reason is that the registration method leads to errors due to the unstable background. Thus, by using mostly texture synthesis our results are sharp but noisy, while VSRS rendering is blurrier. Nevertheless, for S4, “5 → 7”, the proposed method gives better PSNR results than VSRS and VSRS-Alpha-Gist.

In addition to the objective measurements, Fig. 4 shows some subjective results for some sub-frames from S1, S2 and S4 (electronic magnification maybe required). Red-cyan anaglyph images, which are created from two rendered frames beyond the viewing range of the original cameras, are used here to provide a 3-D effect. In Fig. 4 (a)-(d) the sub-frame result from S1 for frame 152 of camera 1 and 3 (using original camera 5 as input) is shown. Fig. 4 (e)-(h) present the sub-frame result from S2 for frame 65 of view 1 and view 3 (using original camera 5 as input) and in Fig. 4 (i)-(l) the sub-frame result from S4 for frame 128 of camera 1 and 2 (using original camera 3 as input) can be seen. The original frames are shown in Fig. 4 (a), (e) and (i). The disocclusion of the warped frames are marked white in Fig. 4 (b) and black in (f) and (j). Fig. 4 (c) and (k) show the results for VSRS and (g) the result for VSRS-Alpha-Gist. Fig. 4 (d), (h) and (l) present the results for the proposed method. As can be seen in Fig. 4 (d), details are well preserved in S1 with our method and structures are reconstructed satisfactorily, while VSRS blends foreground information into the background area (cf. Fig. 4 (c)). For S2 VSRS-Alpha-Gist introduces blur into the unknown area (cf. Fig. 4 (g)) while our method reconstructs the background information in a visually plausible manner (cf. Fig. 4 (h), left beside the column and the back of the person). Fig. 4 (l) shows that structure information is also well preserved at the edge of the image with our method, while VSRS introduces blur into the unknown area (cf. Fig. 4 (k)). Additionally, color blending is prevented by our approach (cf. Fig. 4 (l), left beside the balloon).

8. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a new method to handle the disocclusion problem in DIBR, especially for the extrapolation scenario. Image information from previous and subsequent frames is used in order to perceptibly reduce flickering artifacts in unknown areas. For that, image registration is incorporated into the framework. Only the outermost virtual cameras beyond the viewing range of the original cameras are rendered with a complex approach. All remaining views are interpolated between the original and rendered outermost cameras by using a simpler and faster method. Hence, the overall complexity decreases and spatial consistency between adjacent extrapolated views is maintained, particularly in the synthesized areas. Objective and subjective gains are archived compared to the state-of-the-art.
In future work, further frame patterns for image information prediction from temporally neighboring frames will be evaluated. Advanced image registration methods may also improve the overall performance of the framework.

9. REFERENCES