Consistent Spatio-Temporal Filling of Disocclusions in the Multiview-Video-Plus-Depth Format

Martin Köppel #1, Xi Wang #2, Dimitar Doshkov #3, Thomas Wiegand #45, and Patrick Ndjiki-Nya #6

# Image Communication Chair, Department of Telecommunication Systems, Technical University Berlin
Einsteinufer 17, 10587 Berlin, Germany
1martin.koeppel@tu-berlin.de, 4thomas.wiegand@tu-berlin.de

* Image Processing Department, Fraunhofer Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany
2xi.wang@hhi.fraunhofer.de, 3dimitar.doshkov@hhi.fraunhofer.de,
5thomas.wiegand@hhi.fraunhofer.de, 6patrick.ndjiki-nya@hhi.fraunhofer.de

Abstract—Depth image-based rendering (DIBR) techniques allow for a wide variety of 3-D applications, including synthesizing additional virtual views in a multiview-video-plus-depth (MVD) representation. The MVD format consists of scene texture and depth information for a limited number of original views of the same scene. One of the main obstacles in the DIBR technique lies in the disocclusion problem which results from the fact that a scene can only be observed from a set of original views. This can lead to missing information in the generated virtual views, especially in extrapolation scenarios. Our work describes a novel algorithm that synthesizes such disoccluded textures. The proposed synthesizer enhances the visual experience by taking spatial and temporal video information into account. In order to compensate for global motion in sequences, image registration is incorporated into the framework. Objective and subjective gains are shown compared to three state-of-the-art approaches.

I. INTRODUCTION

Depth image-based rendering (DIBR) techniques have recently become popular for generating additional views in the 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. In order to support autostereoscopic displays as well as to manually fit the depth impression to individual customer habits, the need to calculate additional virtual views arises. Computing these requires the original image points to be reprojected into the 3-D space by using the corresponding depth values. These space points are then projected into the image plane of a virtual camera at the location of the required viewing position. This process is called “3-D image warping” in the computer graphic literature [1]. In the case of a horizontally rectified camera setup, the epipolar lines are horizontally aligned and thus pixel values just need to be shifted along the image rows. The position of the newly projected virtual camera can be between (interpolation) or beside (extrapolation) the viewing range of the original cameras (cf. Fig. 1). One of the most significant problems in DIBR is the question of how to deal with uncovered areas (holes) in the virtual views (cf. Fig. 1), especially in extrapolated views beyond the viewing range of the original cameras. In the extrapolated views, textures which are invisible in all original cameras may become visible. Due to the enhancement of the depth experience through extrapolation in 3D-video, the Moving Pictures Experts Group (MPEG) started experiments to explore the extrapolation capabilities of DIBR algorithms [2]. In the literature, three general methods have been proposed to tackle such holes. First, the depth maps are preprocessed in a way that no disocclusions occur. Usually, the depth map is smoothed, using a symmetric [3] or asymmetric filter [4], to lower gradients. This method gives good results when small baselines need to be compensated. Nevertheless, geometrical distortions can be observed in both foreground and background texture regions. The second way to compensate disocclusions is to cover them with plausible, known image information. Appropriate filling techniques are line-wise filling [5], inpainting methods [6], [7], bilateral filtering [8] and texture synthesis [9]. Alternatively, image domain warping can be utilized to determine virtual views and fill disocclusions by distorting non-salient image regions [10]. A further challenge in DIBR is to maintain temporal consistency in the uncovered areas. First approaches that handle this problem have been published in [9], [11] and [12]. In [9] and [11] a mosaic/sprite is used to store
background information from neighboring frames for further reuse during the filling process. Nevertheless, these approaches are restricted to sequences with static backgrounds. Chen et al. [12] 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. This can yield motion vectors that are different from the real motion. Hence, only little objective and subjective gains are achieved.

In this paper a new approach to handle disocclusions in virtual views is presented. Temporal consistency is achieved by using image information from previous and subsequent frames. By utilizing robust image registration, global background motion between temporally neighboring frames can be compensated. Spatial consistency in virtual views is maintained by first extrapolating and filling the outermost views and then interpolating the further virtual cameras.

II. PROPOSED FRAMEWORK

The proposed framework for DIBR with spatially and temporally consistent texture synthesis is shown in Fig. 2. Initially, the original view and the associated depth map are warped to the outermost left and/or right position (application dependent) beyond the original camera range by using the algorithm proposed in [13] [cf. Fig. 4 (a)]. Then, the depth map is filled at disoccluded locations and a previous and subsequent textured frame is registered to the position of the current frame of the same view by using the image registration method proposed in [14] [cf. Fig. 4 (d)]. Only background texture is utilized to compute the required transformation parameter set. The integration of image registration in order to compensate global camera motion is a new concept in a view synthesis application. Additionally, the hierarchical image registration pattern shown in Fig. 3 and adopted from video coding, is utilized for the first time in a DIBR approach. After the registration step, the holes in the current textured frame are updated with reliable image information from the registered images [cf. Fig. 4 (e)]. Hence, temporal texture consistency in the unknown areas is improved, compared to framewise filling. The holes remaining in the textured frame after this operation are initialized and subsequently filled using texture synthesis [9]. In a final warping step, the remaining virtual views between the outermost filled virtual view and the original views are rendered by applying an extended version of the method published in [13]. Hence, only the outermost views need to be synthesized with a complex approach and the remaining virtual views beyond the original camera range can be interpolated using simpler methods, as disocclusions are typically reflected by small blobs. Hence, spatial consistency between adjacent extrapolated virtual views is maintained.

In the following, the image to be filled is denoted as $F_n$, the associated depth map as $D_{n}$, $F_{np}$ and $F_{nf}$ denote a previous and a subsequent frame (in display order) of $F_n$. Both are registered to $F_n$ and their associated depth maps are $D_{np}$ and $D_{nf}$, respectively. Note that $F_{np}$ and $F_{nf}$ are not necessarily adjacent frames to $F_n$. Holes in a textured image and in a depth map are referred to as $\Omega_F$ and $\Omega_D$, respectively. The original texture in a frame is referred to as $F^o$ with $F^o \subset F\setminus\Omega_F$ and the synthesized texture in a frame is referred to as $F^s$. A pixel position in a textured image or in a depth map is denoted as $p$. $F_i$ refers to the $i$-th frame in a sequence, with $i \in \mathbb{N}$.

III. DEPTH MAP FILLING

The depth map is represented as an 8-bit gray-scale image and indicates the spatial placement of scene objects. This map is used for the image registration step, the frame update procedure and texture filling. The uncovered areas $\Omega_D$, are filled by using the method proposed in [9]. During the depth map filling procedure, an adaptive threshold ($\tau_{min}$) is automatically computed. This depth value can be used to separate foreground and background regions in the image, based on local depth information (for further details please refer to [9]).

IV. IMAGE REGISTRATION PATTERN

Since the outermost views are also used to compute the remaining virtual views (cf. Sec. VI), their quality is of enormous importance for the overall texture filling process. Hence, the outermost textured views are processed with powerful but complex methods. Within a camera view, by using image registration, information from temporally surrounding frames can be used even if the sequence contains global camera motion. Image registration is a process of transforming different sets of data onto one common coordinate system through geometrical mapping. In the utilized image registration method, an area-based model is used to estimate the local affine transformation function, in the luminance channel, with global smoothness. In order to reduce the computational load, a differential multi-scale framework is chosen [14]. Inspired by the work of Schwarz et al. [15], an approach similar to the bi-hierarchical coding structure is employed in order to preserve the temporal...
continuity in the synthesized areas of the virtual views. We use the hierarchical structure to decide which frames to register at the position of the current frame, and from which known image information is obtained to update the current frame.

A. Hierarchical Prediction Structures

A general hierarchical prediction structure has several hierarchical stages [four in [15], three in our framework (cf. Fig. 3)]. In regular intervals, pictures of a sequence are assigned as key pictures (cf. Fig. 3, red frames). A key picture and all pictures that are temporally located between the current key frame and the previous key frame are used to build a group-of-pictures (GOP) in [15]. The pictures of a GOP, except the key picture, are computed using the bi-hierarchical syntax and each of them has two reference pictures [15] (cf. Fig. 3). In Fig. 3 the red frames represent the first, the blue frames the second and the green frames the third hierarchical stage. Note, that in the coding scenario, the key frames have the highest quality of the GOP, i.e. the lower the stage, the lower the reconstruction quality.

B. Proposed Image Registration Pattern

In the proposed framework, the first frame (key frame) of a GOP, is also the last frame of the previous GOP (cf. Fig. 3, red frames). The overlapping GOP structure is used to connect the individual GOPs so that a global consistency among the GOPs is preserved. As shown in Figure 3, the frames that point to the current image to be processed (cf. Fig. 3, processing order) are registered to that position in order to use existing original and/or synthesized image information (cf. Sec. V). By using image information from surrounding pictures to fill the disocclusion, the virtual sequence is temporally stable, especially in the disoccluded areas.

The number of frames in the considered sequence that can be processed with the GOP structure \((N_p, \text{cf. Algorithm 1})\), can be computed as follows:

\[
N_p = N_f - \left( N_f - \left\lfloor \frac{N_f}{G-1} \right\rfloor \cdot G - 1 \right), \quad N_p, N_f \in \mathbb{N}, \tag{1}
\]

where \(N_f\) is the number of frames in the sequence. The remaining frames \((F_{N_p+1}, ..., F_{N_f})\) are filled individually. In general, background textures need to be filled into the disoccluded areas (cf. Fig. 1). Therefore, the global motion should be compensated based on the background areas in an image, i.e. foreground objects must be excluded from the registration step. Based on the depth values in the depth map and the automatically computed threshold \(c_{\min}\) (cf. Sec III), foreground areas/objects can be detected. Hence, only background image information is used to compute the affine transformation matrices [cf. Fig. 4 (c)]. To decide whether the image registration was successful, the PSNR between the current frame and the registered frame is determined for the image region used in the registration step [cf. Fig. 4 (c)]. If the PSNR measured in the luminance channel lies above a chosen threshold \(t_{\text{psnr}}\) the registered frame is considered for updating frame \(F_n\). In a set of objective evaluations, it was found that at least \(t_{\text{reg}} = 80\%\) percent of background information needs to be present to compute the transformation matrix in order to obtain stable and useful registration results. Thus, if necessary \(c_{\min}\) is refined until \(t_{\text{reg}}\) percent of the background information can be used [cf. Fig. 4 (c)].

V. FRAME UPDATE AND TEXTURE FILLING

In this section the update and the synthesis of the outermost view is described. The depth map represents the geometric placement of the objects relatively to the camera position.
Algorithm 1: Pseudo code for GOP update

Data: Frames and associated depth maps.
Result: Synthesized frames.
begin
for $i \leftarrow 1 : 4 : N_p$ do
if $i == 1$ then
F_{i+4} ← Update(F_i, D_i, D_{i+4});
end
else if $i == 4$ then
F_{i-4} ← Update(F_i, D_i, D_{i-4});
end
end

Algorithm 2: Pseudo code for the Update function

Data: $F_n, F_{np}, F_{nf}, D_n, D_{np}$ and $D_{nf}$.
Result: Updated and subsequently synthesized frame $F_n$.
begin
for each pixel position $p \in \Omega_{F_n}$ do
if $F_{np}(p)$ and $F_{nf}(p)$ are in depth range (cf. Eq. 4) then
Compute new texture value (cf. Eq. 5);
else if $F_{np}(p)$ or $F_{nf}(p)$ is in depth range (cf. Eq. 4) then
Choose texture value in depth range;
else
Take no action;
end
end
for each remaining pixel position $p \in \Omega_{F_n}$ do
if $F_{np}(p)$ and $F_{nf}(p)$ are in depth range (cf. Eq. 4) then
Compute new texture value (similar to Eq. 5);
else if $F_{np}(p)$ or $F_{nf}(p)$ is in depth range (cf. Eq. 4) then
Choose texture value in depth range;
else
Take no action;
end
end
Fill remaining holes in $F_n$ using texture synthesis.

Thus, image information with appropriate disparity can be selected during the updating process. The synthesized depth values in $\Omega_D$ (cf. Sec. III) are utilized to decide pixel-wise whether the image information from temporally surrounding registered frames can be used to fill the corresponding hole $\Omega_{F_n}$ in the texture frame. The texture update and filling procedure is shown as pseudo code in Algorithm 1 and 2. Frames are updated with existing original and synthesized data in a GOP-wise manner (cf. Fig. 3, Fig. 4 (e) and Algorithm 1).

The Update function (cf. Algorithm 1 and 2) receives six input images (if all frames are available): $F_n, F_{np}, F_{nf}$, and the associated depth maps $D_n, D_{np}$ and $D_{nf}$. The frame indices $(n, np, nf)$ depend on the current position of $F_n$ in the GOP. Thus, the indices can be computed as follows:

\[ n = i + h, \quad h = [0, G - 1], h \in \mathbb{N}. \]  
\[ np = n - j, \quad nf = n + j, \quad j = [1, G - 1], j \in \mathbb{N}. \]

where $i$ is the position of the frame in the sequence. A maximum of two frames (one previous and one future frame) is considered simultaneously for updating (cf. Fig. 3). Original texture is used primarily and synthesized data as fall-back in the update process. An available pixel position $F_{np}(p)$ or $F_{nf}(p)$ in a registered frame is considered for updating $F_n(p), p \in \Omega_{F_n}$, if the associated registered depth value $D_{np}(p)$ or $D_{nf}(p)$, is in the required depth range (cf. Algorithm 2). This can be formalized as follows:

\[ D_n(p) - t_{dr} < D_{np/nf}(p) < D_n(p) + t_{dr}, \]

where $t_{dr}$ is a selectable parameter to account for small depth variations between temporally adjacent frames. If two values $F_{np}(p)$ and $F_{nf}(p)$ qualify for updating $F_n(p)$ then the following applies (cf. Algorithm 2):

\[ F_n(p) = \frac{F_{np}(p) + F_{nf}(p)}{2}, \quad p \in \Omega_{F_n}. \]

In order, to fill the remaining disocclusions in $F_n$, the method proposed in [9] is used. First, small holes up to a certain threshold (50 pixels as suggested in [9]) are closed by using Laplace PDE [9]. The remaining holes are first initialized and then optimized with patch-based texture synthesis [cf. Fig. 4 (g)]. To account for intensity variations between adjacent patches, covariant cloning is used.

VI. FINAL WARPING

Fig. 5 shows the initial as well as the final warping process. The virtual views beyond the camera range (Fig. 5, white cameras) are interpolated by using the outermost rendered cameras (Fig. 5, gray cameras) and the original cameras (Fig. 5, black cameras) with the improved framework presented in [13]. This method is extended in the way that it can also generate virtual views between an original and a synthesized virtual view. Hence, similar to Sec. IV-A, IV-B and Fig. 3, the consistency among the views is preserved by using two reliable views to compute the inbetween views. Spatially stable results are obtained, especially in the synthesized areas $F^a$. Due to the fact that only the outermost virtual cameras (Fig. 5, gray cameras) need to be rendered with a complex approach in order to achieve visually pleasing results, the overall complexity decreases. Note that the complexity stems from the fact that time-consuming patch-based texture synthesis operations are used for rendering of the outermost views (cf. Sec. V). Remaining virtual views within the original camera baseline are rendered by using [13]. In Fig. 5 three intermediate cameras are interpolated. Nevertheless, depending on the application scenario, the number of interpolated views can be adjusted.
TABLE I: PSNR results by the proposed framework, VSRS, VSRS-Alpha-Etri and VSRS-Alpha-Gist.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Cam.</th>
<th>PSNR(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prop.</td>
<td>VSRS</td>
</tr>
<tr>
<td>S1</td>
<td>5 → 1</td>
<td>41.42</td>
</tr>
<tr>
<td>S1</td>
<td>5 → 9</td>
<td>43.00</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 5</td>
<td>34.61</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 9</td>
<td>33.48</td>
</tr>
<tr>
<td>S3</td>
<td>5 → 5</td>
<td>35.46</td>
</tr>
<tr>
<td>S3</td>
<td>6 → 7</td>
<td>43.24</td>
</tr>
<tr>
<td>S4</td>
<td>5 → 1</td>
<td>34.14</td>
</tr>
<tr>
<td>S4</td>
<td>5 → 7</td>
<td>28.96</td>
</tr>
</tbody>
</table>

TABLE II: SSIM results by the proposed framework, VSRS, VSRS-Alpha-Etri and VSRS-Alpha-Gist.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>Cam.</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prop.</td>
<td>VSRS</td>
</tr>
<tr>
<td>S1</td>
<td>5 → 1</td>
<td>0.9945</td>
</tr>
<tr>
<td>S1</td>
<td>5 → 9</td>
<td>0.9944</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 5</td>
<td>0.9832</td>
</tr>
<tr>
<td>S2</td>
<td>5 → 9</td>
<td>0.9860</td>
</tr>
<tr>
<td>S3</td>
<td>5 → 5</td>
<td>0.7024</td>
</tr>
<tr>
<td>S3</td>
<td>6 → 7</td>
<td>0.9652</td>
</tr>
<tr>
<td>S4</td>
<td>3 → 1</td>
<td>0.9959</td>
</tr>
<tr>
<td>S4</td>
<td>5 → 5</td>
<td>0.4757</td>
</tr>
</tbody>
</table>

**VII. EXPERIMENTAL RESULTS**

The proposed approach is compared to the MPEG view synthesis reference software (VSRS, version 3.5) [13], VSRS-alpha-Etri [7] and VSRS-alpha-Gist [8]. For the evaluation of the proposed algorithm, four MPEG MVD test sequences, which contain global motion are used: “GT-Fly” (S1, 249 frames), “Undo dancer” (S2, 249 frames), “Poznan HallII” (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 evaluate our framework with the following parameter settings: \( t_{\text{psnr}} = 21 \) dB, \( t_{\text{reg}} = 80\% \), \( G = 5 \) and \( t_{\text{dr}} = 15 \). For the texture synthesis method, the parameter settings proposed in [9] are utilized. Only the patch size is changed to 21 due to fact that the resolution of the videos used is higher. We evaluate the extrapolation capabilities of our proposed approach as follows: an outermost virtual view is rendered from an original sequence and its associated depth map, as given in the camera column (Cam.) in Table I. Here, “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 inbetween. We compare our results with those of VSRS, VSRS-alpha-Etri, VSRS-alpha-Gist, which are rendered only from the original camera. The outermost views are evaluated by measuring the PSNR (locally in \( \Omega_P \)) and the SSIM (globally in \( F \)) between the rendered frames and the original data. The interpolated results between the original and the virtual view are evaluated subjectively (cf. Fig. 6).

The objective results given in Table I and II correspond to the mean PSNR and SSIM over all pictures for the outermost rendered sequence. The best result for every sequence among the different methods is highlighted through bold face type. For S1 and S2, the proposed approach performs better than all considered state-of-the-art methods in terms of PSNR and SSIM (cf. Table I and II). For S3 VSRS, VSRS-alpha-Etri and VSRS-alpha-Gist achieve better SSIM results than our method. On the other hand, the proposed method reaches higher PSNR scores for the same sequence. For S4 VSRS and VSRS-alpha-Etri achieve better SSIM results and for S4 “3 → 1” also better PSNR results. The reason for this is that the registration tool does not work reliably for this sequence due to local motion in the background that is different from the global motion. Thus, by using mostly texture synthesis, our results are sharp but noisy, while the rendered results of VSRS and VSRS-alpha-Etri are blurrier. Nevertheless, we reached some objective gains in terms of PSNR for S4 “5 → 7” as well as some subjective improvements (cf. Fig. 6 (i)-(l)).

In addition to the objective measurements, Fig. 6 shows subjective results for some sub-frames from the sequences S1, S2 and S4 (electronican magnification maybe required). Red-cyan anaglyph images, which are created from two rendered frames beyond the viewing range of the original cameras, are shown here to provide a 3-D effect. Fig. 6 (a)-(d) show the sub-frame results for frame 155 of view 1 and view 3 for S1 (rendered from original view 5). Fig. 6 (e)-(h) show the sub-frame results from frame 56 of view 1 and view 3 for S2 (rendered from original view 5) and Fig. 6 (i)-(l) present the sub-frame results from frame 42 of view 1 and view 2 for S4 (rendered from original view 3). The original data is depicted in Fig. 6 (a), (e) and (i). The dissocclusions are marked white in Fig. 6 (b) or black in (f) and (j). Fig. 6 (c), (g), (k) show the results for VSRS-alpha-Gist, VSRS-alpha-Etri and VSRS, respectively. The outcomes of the proposed method are shown in Fig. 6 (d), (h) and (l). As can be seen in Fig. 6 (d), details are well preserved with our method and structures are reconstructed satisfactorily, while VSRS-alpha-Gist generate color artifacts and smoothen the synthesized textures [cf. Fig. 6 (c)]. VSRS-alpha-Etri fills the holes with foreground and background texture [cf. Fig. 6 (g)] while our method synthesizes satisfactorily, while VSRS-alpha-Gist generate color artifacts and smoothen the synthesized textures [cf. Fig. 6 (c)]. VSRS-alpha-Etri fills the holes with foreground and background texture [cf. Fig. 6 (g)] while our method synthesizes the missing regions only from the background [Fig. 6 (h)]. Furthermore, structure information is also well preserved with the new method (cf. Fig. 6 (h), left, beside the arm and the column). It can be seen in Fig. 6 (k) VSRS introduces blur into the uncovered areas, while the proposed method synthesizes sharp textures (cf. Fig. 6 (l), left beside the balloons and in the edge of the image).

**VIII. 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 artifacts in unknown areas. To compensate the global motion in a sequence, image registration is incorporated into the framework. Only
Fig. 6: Sub-frame results for the sequences “GT-Fly” (frame 155), “Undo dancer” (frame 56) and “Balloons” (frame 42).

(a), (e), (i) Original. Warped image with the disoccluded areas marked white (b) or black (f), (j). Final results using (c) VSRS-alpha-Gist, (g) VSRS-alpha-Etri, (k) VSRS and (d),(h),(l) the proposed approach. Please use red-cyan glasses.

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 using a simpler and faster method. Hence, the overall complexity is minimized and spatial consistency between adjacent extrapolated views is maintained, particularly in the synthesized areas.

ACKNOWLEDGEMENT

The authors would like to thank the Poznan University of Technology, Nokia and the Nagoya University for providing the “Poznan Hall2”, “GT-Fly”, “Undo Dancer” and “Balloons” sequences, respectively. This work was funded in part by the German Research Foundation (DFG).

REFERENCES