Depth Image-Based Rendering With Advanced Texture Synthesis for 3-D Video

Patrick Ndjiki-Nya, *Member, IEEE*, Martin Köppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, *Student Member, IEEE*, Karsten Müller, *Senior Member, IEEE*, and Thomas Wiegand, *Fellow, IEEE*

***Abstract*—A depth image-based rendering (DIBR) approach with advanced inpainting methods is presented. The DIBR algorithm can be used in 3-D video applications to synthesize a number of different perspectives of the same scene, e.g., from a multiview-video-plus-depth (MVD) representation. This MVD format consists of video and depth sequences for a limited number of original camera views of the same natural scene. Here, DIBR methods allow the computation of additional new views. An inherent problem of the view synthesis concept is the fact that image information which is occluded in the original views may become visible, especially in extrapolated views beyond the viewing range of the original cameras. The presented algorithm synthesizes these occluded textures. The synthesizer achieves visually satisfying results by taking spatial and temporal consistency measures into account. Detailed experiments show significant objective and subjective gains of the proposed method in comparison to the state-of-the-art methods.**

***Index Terms*—Depth image based rendering, inpainting, texture synthesis, view synthesis, 3-D video.**

I. Introduction

THREE-DIMENSIONAL (3-D) video is rapidly growing in popularity as many stereo video products are currently entering the mass market. The viewing of stereo video on a multi-user stereo display requires the use of additional eyeglasses. Autostereoscopic multiview displays (we call them 3-D displays) provide 3-D depth perception without the need to wear additional eyeglasses by showing a number of slightly different views simultaneously. The various views for the 3-D display are typically generated from fewer images captured by original cameras at different viewpoints and separated by a baseline corresponding to the human eye distance. 3-D displays ensure that a viewer always sees a stereo pair from predefined viewpoints. The high number of views required by 3-D displays poses a challenge to the acquisition as well as to the transmission process, as only a limited number of original views can be recorded, stored and transmitted. Consequently, the need to render additional virtual views from transmitted views arises, in order to support 3-D displays.

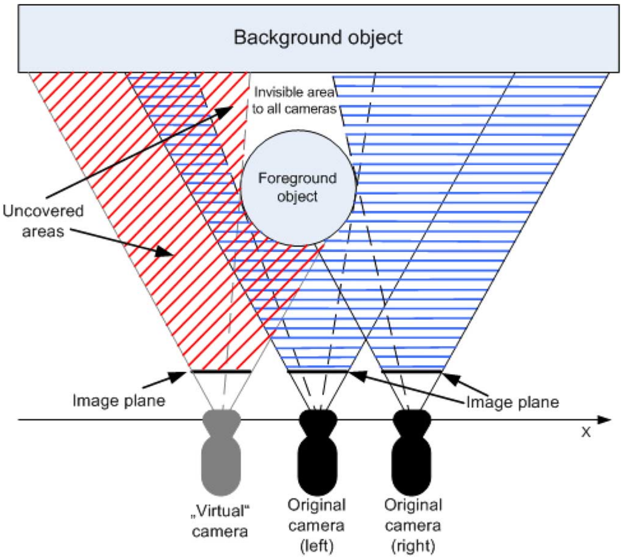


Fig. 1. Extrapolation scenario of a virtual camera view from original cameras: scene areas that are visible in the viewing cones of the original cameras are marked with horizontal lines. Those visible in the virtual view are highlighted by diagonal lines. Uncovered areas are visible in the virtual camera but invisible in the original ones and must thus be reconstructed synthetically.

The basic principle for generating new views is projective geometry, where the 3-D geometry of the original scene as well as the camera locations, e.g., in the form of extrinsic and intrinsic camera parameters, must be known. Given the depth information included in MVD data, mapping of image contents into the 3-D scene space and subsequently into the virtual camera view can be conducted [1]. For horizontally rectified camera setups, the projection is simplified as disparities between samples in available views are reduced to the horizontal direction, i.e., to the same row in the images [1].

Depth image-based rendering (DIBR) is a technology for synthesizing novel realistic images at a slightly different view perspective, using a textured image and its associated depth map. A critical problem is that the regions occluded by the foreground (FG) objects in the original views may become visible in the synthesized views. This is particularly problematic in case of extrapolation beyond the baseline of the original views, as shown in Fig. 1. Disocclusions are typically less critical in view interpolation as uncovered textures in background regions and at

Manuscript received November 01, 2010; revised February 28, 2011; accepted March 08, 2011. Date of publication March 17, 2011; date of current version May 18, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Homer H. Chen.

P. Ndjiki-Nya, M. Köppel, D. Doshkov, H. Lakshman, P. Merkle and K. Müller are with the Image Processing Department, Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute (HHI), 10587 Berlin, Germany (e-mail: patrick.ndjiki-nya@hhi.fraunhofer.de; martin.köppel@hhi.fraunhofer.de; dimitar.doshkov@hhi.fraunhofer.de; haricharan.lakshman@hhi.fraunhofer.de; philipp.merkle@hhi.fraunhofer.de; karsten.mueller@hhi.fraunhofer.de).

T. Wiegand is with the Image Processing Department, Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute (HHI), 10587 Berlin, Germany, and also with the Department of Telecommunication Systems, School of Electrical Engineering and Computer Science, Berlin Institute of Technology, 10587 Berlin, Germany (e-mail: twiegand@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2128862

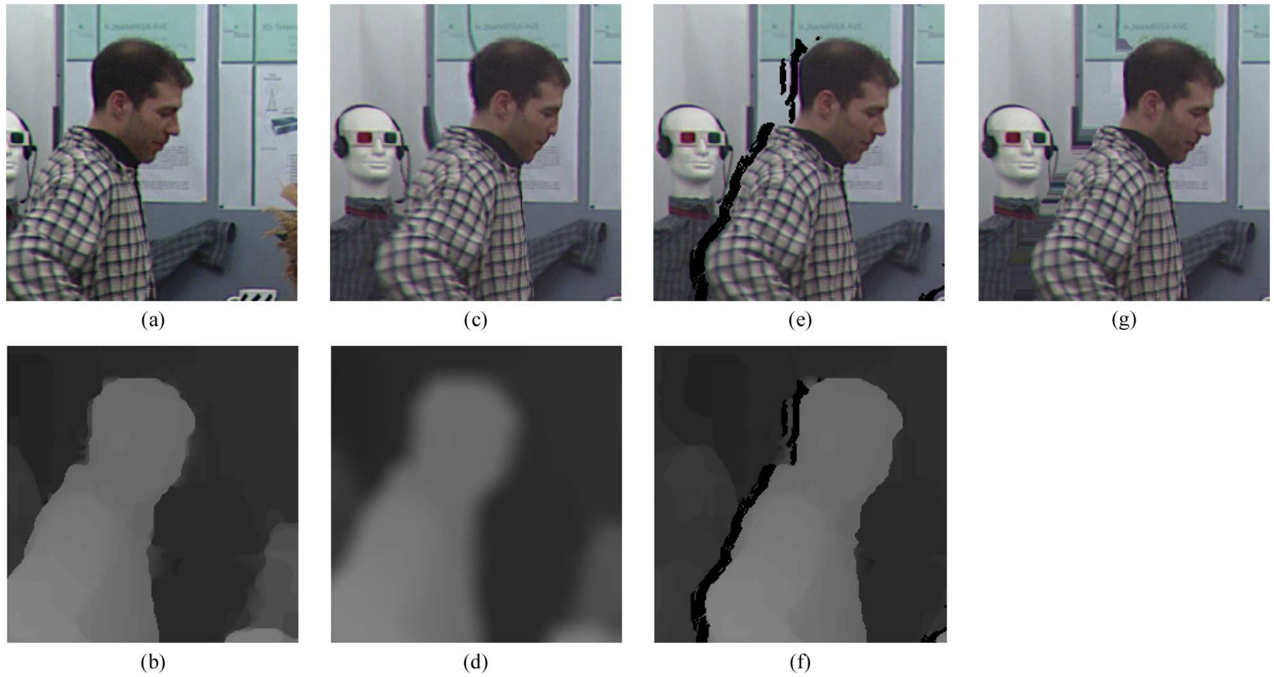


Fig. 2. DIBR results for frame 95 of the "Book Arrival" sequence. A baseline of 130 mm is used. (a) Original view. (b) Corresponding depth map. (c) Virtual camera view obtained from (d) the Gaussian filtered depth map. (e) Warped virtual view generated based on the original depth map in (b). (f) Corresponding warped original depth map with disocclusions (marked black). (g) Result of line-wise filling approach (see artifacts at the person's back).

frame boundaries are often visible in one of the original cameras [2]–[5].

In the literature, two basic options are described to address the disocclusion problem. Either the missing image regions are replaced by plausible color information [20] or the depth map is preprocessed in a way that no disocclusions appear in the rendered image [21]. The latter technique will be referred to as "disocclusion elimination" in the following.

The color filling methods process uncovered regions by classifying the samples in their vicinity into foreground and background. The known background samples are then adequately inserted into the disoccluded region [2], [4], [5], [9]. In [9], Mark applies interpolation, while Zinger *et al.* [4] and Mori *et al.* [5] use a simple inpainting algorithm [6] to solve this problem. However, interpolation and simple inpainting methods tend to introduce blur in the unknown areas. Another simple approach repeats the last valid background sample line-wise into the unknown area [8]. Filling methods based on this approach suffer from severe artifacts when structured backgrounds and dominant vertical edges are present [9] [cf. Fig. 2(g)]. A more advanced algorithm was proposed by Jiufei *et al.* [10]. They fill the uncovered area via texture synthesis and are able to show that better visual results can be achieved compared to the previously described methods. Unfortunately, neither postprocessing of the selected continuation patches nor of the uncovered areas is done, which may result in block artifacts or luminance inconsistencies in the virtual view.

Disocclusion elimination methods usually apply a lowpass filter to preprocess depth maps. A Gaussian low-pass filter [21] [see Fig. 2(d)] or, alternatively, an asymmetric filter [12], [13] is often used. This depth prefiltering step corresponds to smoothing depth data across the edges and thus lowering the depth gradients in the virtual view. As highlighted by the example in Fig. 2(c), foreground objects can be considerably distorted by this approach, which is subjectively quite disturbing [11]–[13], [21].

Combinations of color filling and disocclusion elimination approaches can be found in the literature. Cheng *et al.* [14] preprocess the depth map with a bilateral filter and then fill the uncovered areas in the corresponding textured image using available background texture information. Chen *et al.* [11] preprocess the depth map with an edge-dependent Gaussian filter and fill the remaining holes via edge-oriented interpolation. Here, typical artifacts of both approaches can be observed, albeit at a reduced visual impact.

Another disadvantage of existing solutions is that they support only small baselines for rendering and yield very annoying artifacts for larger baseline rendering. As displays with 50 views and more are expected in a few years, larger baselines and consequently visually smooth texture synthesis of larger disocclusions are required. Furthermore, all approaches considered so far render the new images frame by frame, ignoring the temporal correlation of the filled areas, and therefore causing typical flickering artifacts in the virtual view. First approaches for handling temporal consistency in uncovered regions have been published recently [15], [16]. Schmeing and Jiang [15] first determine the background information using a background subtraction method. Based on that, the uncovered areas are filled. Two major drawbacks of this approach are caused by the omission of illumination variation compensation and the required manual correction of disocclusions.

Chen *et al.* [16] assume that the original views are H.264/AVC [17] encoded and therefore extract their motion vectors directly from the bit stream. These are used to compensate known locations in a virtual frame based on temporal neighbors in the virtual view. However, the determination of motion vectors in an H.264/AVC encoder is steered by the overall encoder optimization used, yielding motion vectors that can be rather different from the real motion. The benefit of using these motion vectors is additionally limited by the fact that they are sparsely sampled and thus lack accuracy [18]. Hence, the PSNR and SSIM [19] gains presented in the work by Chen *et al.* [16] are very small compared with the state-of-the-art methodology.

As mentioned before [6], [10], texture synthesis [22]–[30] is an appropriate technique for filling missing image regions with known information. This method operates in parametric [23]–[25] or nonparametric [22], [26]–[30] modes. Parametric synthesis approaches generate new textures using a compact model with a fixed [23], [24] or dynamic [25] parameter set. Nonparametric approaches, on the other hand, typically formulate the texture synthesis problem based on the Markov random field (MRF) theory [27]–[30]. Nonparametric approaches can be further classified in sample- or patch-based methods. Sample-based algorithms update the synthetic texture sample-wise [27], while patch-based approaches apply a patch-wise update [28]–[30], i.e., a set of samples is updated simultaneously. Typically nonparametric synthesis approaches yield better inpainting results than parametric algorithms, and in addition they can also be successfully applied to a much larger variety of textures [29]. For the restoration of small and rather homogenous regions, inpainting approaches are often used that are based on solving partial differential equations (PDEs) [7].

In this paper, a new approach for handling disocclusions in synthesized views for 3-D video is presented. The method is based on nonparametric texture synthesis. Statistical dependencies between different pictures of a sequence are taken into consideration via a background (BG) sprite. A robust initialization gives an estimate of the unknown image regions that is refined during the synthesis stage.

The remainder of this paper is organized as follows. The overall algorithm is presented in Section II. Depth map filling, image initialization, and texture synthesis are presented in detail in Sections III–VI. In Section VII, the experimental results are presented. Finally, conclusions and future steps are given in Section VIII.

II. Proposed View Synthesis Framework

The proposed framework for virtual view generation with time-consistent texture synthesis is outlined in Fig. 3. The texture images and associated depth maps (DMs) of an MVD sequence are taken as input. DMs are provided with the test data, e.g., by depth estimation methods by Tanimoto *et al.* in [31]. Next, a projection of the original views towards the virtual views based on the DMs is realized with an algorithm similar to [32], where the foreground (FG) and background (BG) objects are warped separately. This results in an image with holes from disoccluded background, as shown in Fig. 2(e).

In addition, the DM is also projected [see Fig. 2(f)] for later foreground–background separation in the texture synthesis stage. According to the original scene capturing setup, background motion in all views may occur. In this case, a motion estimation stage needs to be included in the workflow after both projection steps (see Fig. 3) to compensate for the global BG motion. In our simulations, we first concentrated on a static BG similarly to the work of Schmeing and Jiang [15]. Note, however, that our algorithm is fully automatic, i.e., no manual disocclusion correction is required, and can seamlessly compensate illumination changes.

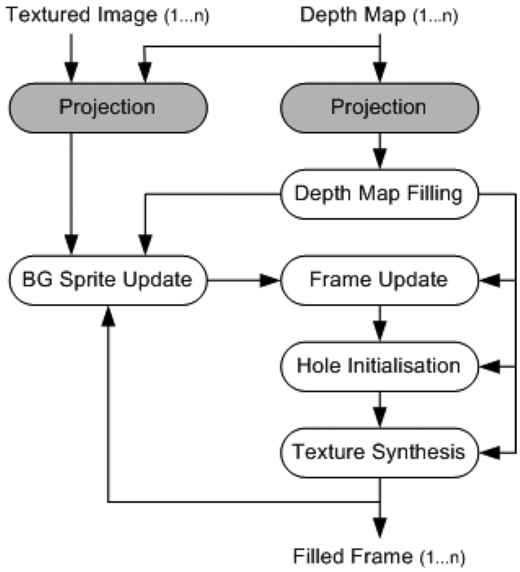


Fig. 3. Block diagram of the proposed approach. First, disocclusions in the DM are filled after the projection step by Tanimoto *et al.* [32]. Next, the BG sprite is updated with original BG data and the holes in the current picture are updated from the BG sprite. Then, the remaining holes are initialized and refined with texture synthesis. Finally, the BG sprite is updated with the new synthesized texture.

The goal of the new view synthesis algorithm is to fill the disocclusions (holes) resulting from the warping process. They become visible both in the virtual DM and the textured image and must be filled in a visually plausible manner. For video sequences, this includes a temporally stable synthesis result, meaning that information from temporally neighboring frames should be taken into account. For minimizing the processing delay, only causal neighbors are considered in this work. Temporal consistency is achieved with a BG sprite, which stores background information from processed frames. In a first step, the disoccluded areas in the DM are filled as shown in the next section. The BG sprite is then updated with known BG information from the current picture. Next, the holes in the current picture are updated from the BG sprite. The remaining holes are treated by first initializing the area from spatially adjacent original texture, providing an estimate of the missing information. In the next step, patch-based texture synthesis is used to refine the initialized area. The BG sprite is finally updated with the synthesized image information for temporal consistency during the filling of holes in the subsequent pictures.

III. Filling Disocclusions in the Depth Map

Given the inherent properties of the depth-based image warping, larger uncovered areas mostly belong to BG objects. The DM is represented as an 8-b grayscale image, denoted as D in the following. The continuous depth range of the scene is quantized to the discrete depth values, assigning the value 255 to the point that is closest to the camera and 1 to the most distant point [Fig. 4(a)–(c)]. The holes in the DM are

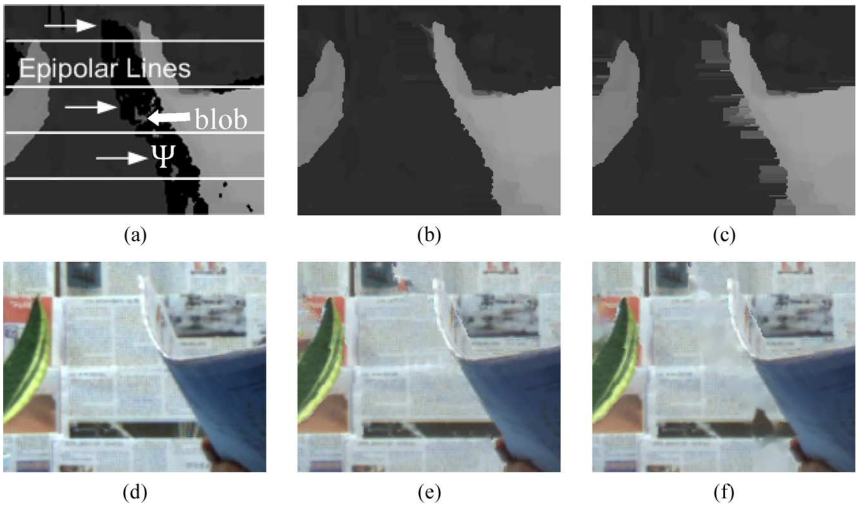


Fig. 4. Results for picture 1 of the "Newspaper" sequence for the proposed depth map and texture filling approach. (a) Depth map with disoccluded area marked black (filling direction given by white arrows). (b) Result of proposed depth map filling approach. (c) Line-wise filling of depth map without blob removal. (d) Original reference image. (e) Result of the proposed approach. (f) Result of MPEG VSRS.

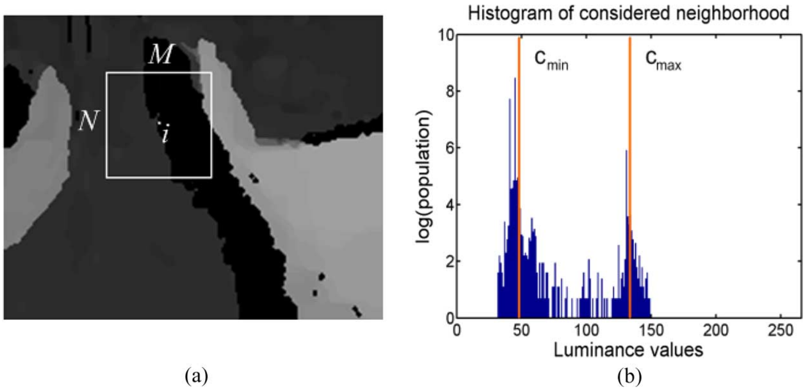


Fig. 5. (a) Depth map with highlighted neighborhood (square) centered at i. (b) Histogram of considered neighborhood with the two centroids $\mathrm{c}^i_{\min}$ and $\mathrm{c}^i_{\max}$, clustered via k-means.

currently assigned the value 0. In Fig. 4(a), the uncovered area in the DM is denoted as Ψ and the corresponding boundary is denoted as $\partial\Psi$. $\partial\Psi$ thereby corresponds to the outer boundary of Ψ and consists of known background depth values. Due to inaccuracies in depth estimation, FG object boundary samples may be warped into Ψ [denoted as "blobs" in the following, Fig. 4(a)]. One possibility to proceed is to fill the last known BG depth value D_i, with $i \in \partial\Psi$ line-wise into Ψ, as proposed in [8] [Fig. 4(c)].

In this work, small blobs up to γ samples in Ψ are assigned to Ψ, as they are assumed to correspond to noise and may otherwise lead to noticeable inaccuracies in the postprocessed DM [see Fig. 4(b) and (c)]. Subsequently, a verified D_i value is copied line-wise into Ψ. It is assumed that relying on a single value of D_i can be error-prone. Hence, the spatial neighborhood surrounding location i is clustered into two depth classes, whose centroids are represented by $c^i_{\min}$ and $c^i_{\max}$. They represent FG and BG depth values respectively (see Fig. 5). The neighborhood I, is given by a rectangular area of $M \times N$ samples $(M, N \in \mathbb{N})$ and centered at location i. $c^i_{\min}$ and $c^i_{\max}$ are computed via k-means clustering $(k = 2)$ [33]. After $c^i_{\min}$ and $c^i_{\max}$ estimation, depth information at locations $j \in \Psi$ and on the same row as i is extrapolated. The selection criterion for the depth values to be filled at locations j is defined as follows:

$$D_{\mathrm{j}} = \begin{cases} D_i, & \text{if } D_i \leq c^i_{\min} \\ c^i_{\min}, & \text{otherwise} \end{cases}, \quad j \in \Psi \wedge j_y = i_y; i \in \partial\Psi \quad (1)$$

where j_y and i_y correspond to the row coordinates of locations j and i, respectively. Background–foreground clustering and subsequent line-wise filling is done for all $i \in \partial\Psi$.

So far, we have increased the robustness to artifacts in depth-map filling. The computed $\{\mathrm{c}^i_{\min}\}$ values are stored in order to be used for image and sprite updating as explained in the next section.

IV. Sprite and Image Updating

The BG information and its associated depth values are stored as a BG sprite, denoted as S [cf. Figs. 13(c) and 14(c)] and DM sprite, denoted as G [cf. Figs. 13(d) and 14(d)]. These sprites accumulate valuable information for rendering textured images. In fact, by referencing the sprite samples for filling unknown area in the current picture, the synthesis is temporally stabilized.

A. Sprite Update

For each new picture, denoted as P, the depth values of all sample positions $l \in D \setminus \Psi$ are examined to determine the samples that can be considered for sprite update. For that, the following content-adaptive threshold is required:

$$\overline{c_{\min}} = \begin{cases} c^i_{\min}\left(\frac{|\partial\Psi|+1}{2}\right), & \text{if } |\partial\Psi| \text{ is odd} \\ \frac{1}{2}\left[c^i_{\min}\left(\frac{|\partial\Psi|}{2}\right) + c^i_{\min}\left(\frac{|\partial\Psi|}{2}+1\right)\right], & \text{if } |\partial\Psi| \text{ is even} \end{cases} \qquad (2)$$

where $\overline{c_{\min}}$ is the median value of the sorted $c^i_{\min}$ values denoted as $c^i_{\min}(1)\ldots c^i_{\min}(|\partial\Psi|)$. Hence, all samples with a depth value below $\overline{c_{\min}}$ are eligible for sprite update.

Depth values below $\overline{c_{\min}}$ are assumed to describe the BG, while the remaining values are assigned to the FG. Due to the mentioned inaccuracies in the depth estimation step, depth estimates along background–foreground transitions and within the uncovered area in P, denoted as Ω, are considered as being unreliable. Therefore, a two-sample-wide area around the unreliable regions is not considered for sprite update. The remaining locations with $D_l < \overline{c_{\min}}$ are stored in the BG sprite S, and DM sprite G, respectively, where previously assigned color or depth information is overwritten in S and G. After the synthesis step (cf. Sections V and VI), newly synthesized textures and depths are incorporated into the sprites as well.

B. Textured Image Update

The disoccluded regions of every picture P are updated from the BG sprite S. Sample positions corresponding to samples in the BG sprite with unknown background information are ignored. The sample positions in S to be used for the update of the current picture P are selected as follows:

$$P_m = \begin{cases} S_m, & \text{if } D_m < G_m + \beta \\ P_m, & \text{otherwise} \end{cases} \quad \forall\, m \in \Omega \qquad (3)$$

where P_m, S_m represent the intensity value at location m in the current picture and the BG sprite, respectively. D_m and G_m represent the depth value at location m in the extrapolated DM and the DM sprite, respectively. The parameter β allows for some variance in the local BG depth value. The disoccluded area in P is denoted as Ω. Note that (3) is applied to the chroma channels in the same way.

In order to account for illumination variations, the covariant cloning method [30], [34], [35] is utilized to fit the BG sprite samples to the intensity distribution in the relevant neighborhood of the current picture. The term "cloning" or "seamless cloning" denotes the process of replacing a region of a given picture by another content (often from a different picture), such that subjective impairments are minimized. In [38] Poisson cloning is used in texture synthesis to reduce the photometric seams in the gradient domain.

In order to explain the cloning principle, we define f^* as a known scalar function over the domain P ($P \subset \mathbb{R}^2$). As indicated in Fig. 6(a), $\partial\Omega$ represents the boundary of the unknown area $\Omega \subset P$. g is a function defined over the texture source R to be (partially) mapped onto Ω. f is an unknown scalar function defined over Ω.

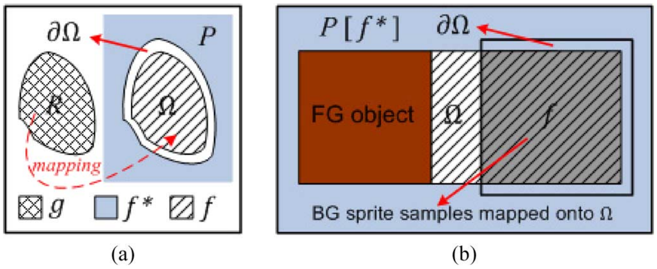


Fig. 6. (a) Seamless cloning principle. (b) Cloning application to the view synthesis framework.

The aim is to find f using the source function g and the information available in $\partial\Omega$. This boundary value problem can be expressed as [34]

$$\Delta f = \Delta g \frac{f}{g} \qquad (4)$$

with the Dirichlet boundary condition

$$f|_{\partial\Omega} = f^*|_{\partial\Omega} \qquad (5)$$

where Δ represents the Laplacian operator. In this way, information on the boundary $\partial\Omega$ is diffused into Ω, such that the transition between the source function g and P is smooth.

The notations of covariant cloning, in the context of the proposed view synthesis framework, are illustrated in Fig. 6(b). It can be seen that boundaries of the area covered by the BG sprite, when mapped onto Ω in the current picture P, are either adjacent to the nonreconstructed area Ω or directly to the FG object [not shown in Fig. 6(b)]. In this case the BG sprite area is represented by g. The cloned BG sprite area then corresponds to f. As can be seen in Fig. 6(b), Ω may remain partially unknown. The current image is denoted as f^* and the boundary $\partial\Omega$ comprises the spatial neighbors of the BG sprite samples. Due to the presence of uncovered areas (FG objects), the boundary conditions (5) for the region Ω are incomplete, i.e., $\partial\Omega$ is undefined at these edges. Therefore, the cloning method is adapted to the given view synthesis framework by ensuring that only BG samples in the current picture are considered as valid boundary conditions:

$$\begin{cases} f|_{\partial\Omega} = f^*|_{\partial\Omega}, & \text{default} \\ f|_{\partial\Omega} = g|_{\partial\Omega}, & \text{if } \partial\Omega \text{ undefined.} \end{cases} \qquad (6)$$

This modified boundary condition implies that the color information is only diffused into the BG sprite samples from those boundaries for which $\partial\Omega$ is defined. This diffusion process is also called photometric correction. Please note that for simplifying the cloning approach, the quotient f/g from (4) is approximated by a constant. We set $f/g = 1$, which transforms (4) exactly to the corresponding one in the work by Pérez *et al.* [36].

V. Initialization of Textured Images

The disocclusions remaining after sprite and image updating are preprocessed through a new texture initialization algorithm. In a first step, the Laplacian equation [36] is used to fill small holes in the current image [Fig. 7(a) and (c)]. For the reconstruction of smooth regions this method gives satisfactory re-

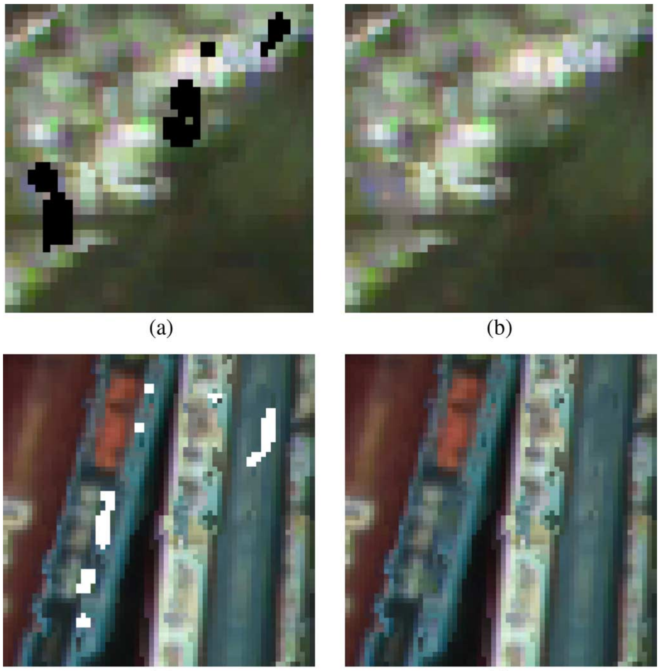


Fig. 7. Results for hole filling with Laplacian cloning. Disoccluded areas: (a) black or (c) white. (b), (d) Filled disocclusions.

sults [Fig. 7(b) and (d)]. Good visual results are observed for holes smaller than γ samples (e.g., 50 samples), where Laplace cloning is about 10 times faster than patch-based texture synthesis (cf. Section VI). Hence, after Laplace cloning, small holes are regarded as finally filled and are not considered in the texture refinement step.

For holes larger than γ samples, we have shown in our previous work [37] that the visual results of texture synthesis can be improved by using an initial estimate of sample values. In this paper, we present an initialization method that is based on the statistical properties of known samples in the vicinity of Ω. Generally, the known samples constitute valid BG samples, but in some cases the depth values at the foreground–background transition are not reliable. Hence, the probability distribution of known BG sample values in the spatial neighborhood of the hole area is observed to be skewed.

In order to determine the BG value from spatially adjacent samples, the median estimator is used, which is the standard measure of (end value) location used in case of skewed distributions. A window of samples sized 32 × 32 and centered around the sample to be filled is considered. For each unknown sample, a measure N_{BG} is set equal to the number of known samples that are classified as BG in the current window. The unknown samples are visited in decreasing order of N_{BG}. A 2-D median filter operates on the BG samples in the current window and the filtered output is used to initialize the unknown sample. The filtering operation can be viewed as the process of extracting a valid BG value from the spatially neighboring samples. This serves as a coarse estimate that can be used at the texture synthesis stage to recover the details in the unknown region. Using the described initialization scheme, the sensitivity of the texture synthesis stage to outliers is significantly reduced.

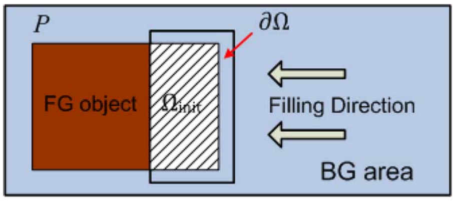


Fig. 8. Hole-filling order from background data at the texture refinement step.

VI. Texture Refinement via Synthesis

In texture synthesis techniques the unknown region is synthesized by copying content from the known parts $(P - \Omega)$ to the missing parts (Ω) of the image. Patch-based texture synthesis is used in this work to refine the initialized areas. The patch filling order criterion introduced by Criminisi *et al.* [22] is utilized. They determine the current filling position at $\partial\Omega$ via a priority term. The priority is the product of the confidence term and the data term. The confidence term enforces a concentric filling order, while the data term encourages linear structures to be synthesized first. Their approach is enhanced in two ways in this work. First, the gradient is calculated for the original as well as the initialized samples (Fig. 8, Ω_{init}). This leads to a better isophote direction compared to [22]. Second, the filling order is steered such that the synthesis starts from the BG area towards the FG objects. For steering the filling direction, only the border sample positions located in the BG are assigned filling priorities according to [22] (Fig. 8, $\partial\Omega$ sample positions). In the following, the patch at the current location to be filled is denoted as $\boldsymbol{c}$. Its center is denoted as $\boldsymbol{c}_{\text{center}}$. An area around $\boldsymbol{c}_{\text{center}}$ is defined to be the source area A. The filling algorithm now searches for a patch $\boldsymbol{x}$ of size $L \times Q$, centered at $\boldsymbol{x}_{\text{center}}$, in A that is similar to $\boldsymbol{c}$.

In the matching routine, only the luminance channel is considered. Given the filled DM, the depth value of $\boldsymbol{c}_{\text{center}}$ is always known. All sample positions in A with depth values higher than $D_{\text{center}} + \beta$ are excluded from the source area, that is, they are not considered as center, $\boldsymbol{x}_{\text{center}}$, of the continuation patch. Therefore, the likelihood of selecting patches with depth values much higher than the current region to be filled is reduced. To speed up the matching procedure, the source area is subsampled by a factor s. The remaining source positions are used as center positions of $\boldsymbol{x}$. The best continuation patch out of all candidate patches in the source area is obtained by minimizing the following cost function:

$$E = \sum_{i=1}^{K} \|x_i - c_i\|^2 + w_\Omega \sum_{j=1}^{K_\Omega} \|x_j - c_j\|^2 \tag{7}$$

where K is the number of original and K_Ω is the number of initialized samples in $\boldsymbol{c}$. w_Ω is the weighting factor for the initialized values in Ω. To ensure smooth transitions between adjacent patches, an efficient postprocessing method, based on covariant cloning [30] and similar to the photometric correction method described in Section IV-B, is utilized. This postprocessing approach is adapted to the framework in such a manner that FG objects are not considered as boundary samples.

VII. Experimental Results

Here, detailed experiments are described. In Section VII-A, the data set as well as the evaluation measures used are defined. In Sections VII-B, VII-C, and VII-D, relevant modules of our view synthesis framework, i.e., the depth-map filling algorithm, the texture synthesis method, and the sprite updating module, are evaluated to assess their contribution to the overall system's performance.

A. Data Set and Quality Measures

For evaluating the proposed algorithm, four MVD test sequences are used: "Book arrival" (S1, 100 frames), "Lovebird1" (S2, 150 frames), "Newspaper" (S3, 200 frames), and "Mobile" (S4, 200 frames). S1, S2, and S3 have a resolution of 1024 × 768 samples, while S4 has a resolution of 720 × 540 samples. For each sequence, the rectified videos of several views with slightly different camera perspectives are available. The baseline between two adjacent cameras is approximately 65 mm for all test sequences. We consider one or two original—but not necessarily adjacent—cameras (left and right view) to assess the performance of our approach. The following two scenarios are evaluated:

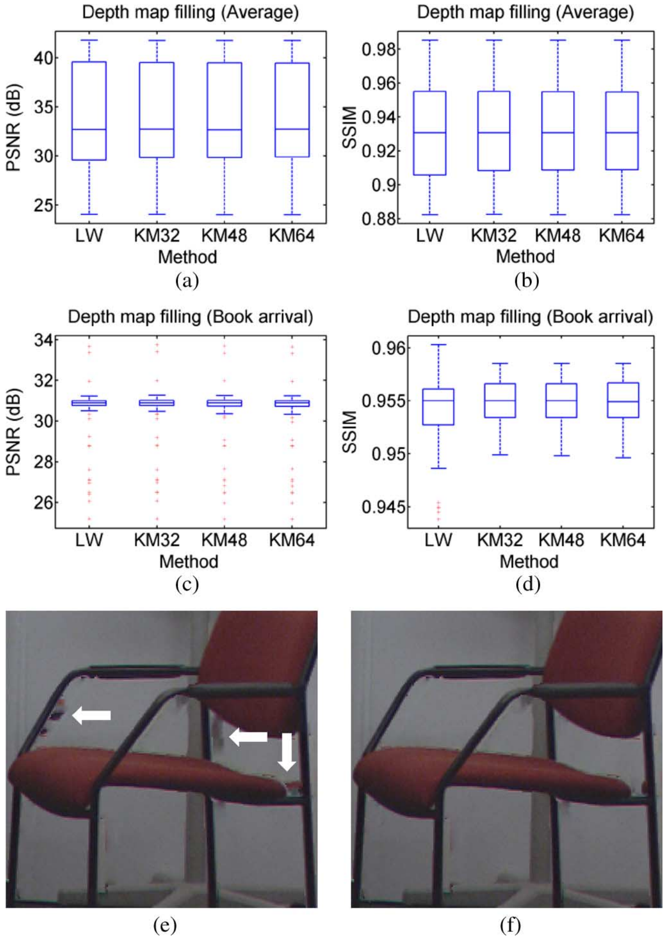
- view synthesis with the regular baseline of adjacent cameras (∼65 mm);
- view synthesis with twice the regular baseline of adjacent cameras (∼130 mm).

The performance of the proposed view synthesis algorithm is assessed with PSNR and SSIM. For the presented results, PSNR is computed locally, that is, only for the defective area in the image, while SSIM is determined for the entire image as it cannot be easily applied to arbitrary shaped regions. SSIM is provided in addition to PSNR, since the use of PSNR is difficult in case of geometric distortions as they often occur in synthesized images [19].

B. Assessment of the Depth Map Filling Algorithm

As mentioned in Section III, the most important DM filling parameter is the k-means clustering window sized $M \times N$. Experiments were conducted for all video sequences assuming a square window, i.e., $M = N$. Furthermore, all tests were performed using twice the regular baseline in order to have larger disoccluded areas. Note that the PSNR and SSIM results are measured using the textured images. Fig. 9(a), (b) depict the average values that were achieved for PSNR and SSIM over the whole test set. It can be seen that all filling methods (line-wise without blob removal (LW) and k-means clustering with different window sizes (32 × 32, 48 × 48, 64 × 64)) perform similarly. Note that line-wise filling without blob removal gives slightly better objective results than line-wise filling with blob removal.

Comparable results are observed for the "Book arrival" (S1) sequence as highlighted in Fig. 9(c) and (d). Therefore, subjective results are taken into consideration to find the optimal filling method. As shown in Fig. 9(e) and (f), for "Book arrival" (S1), distortions can be observed for the LW approach, while k-means clustering generates good results. Similar results are observed for all other test sequences. Extensive viewing of the test sequences leads us to the conclusion that the k-means clustering method with window size $M = N = 32$ produces the best results.



(a) (b) (c) (d) (e) (f)

Fig. 9. Influence of the proposed depth-map filling method on the filling results. Average values for (a) PSNR and (b) SSIM, over the whole test set. Objective results for the sequence "Book arrival" with (c) PSNR and (d) SSIM. Subjective differences between (e) line-wise method without blob removal (LW) and (f) k-means clustering with $M = N = 32$ (KM32).

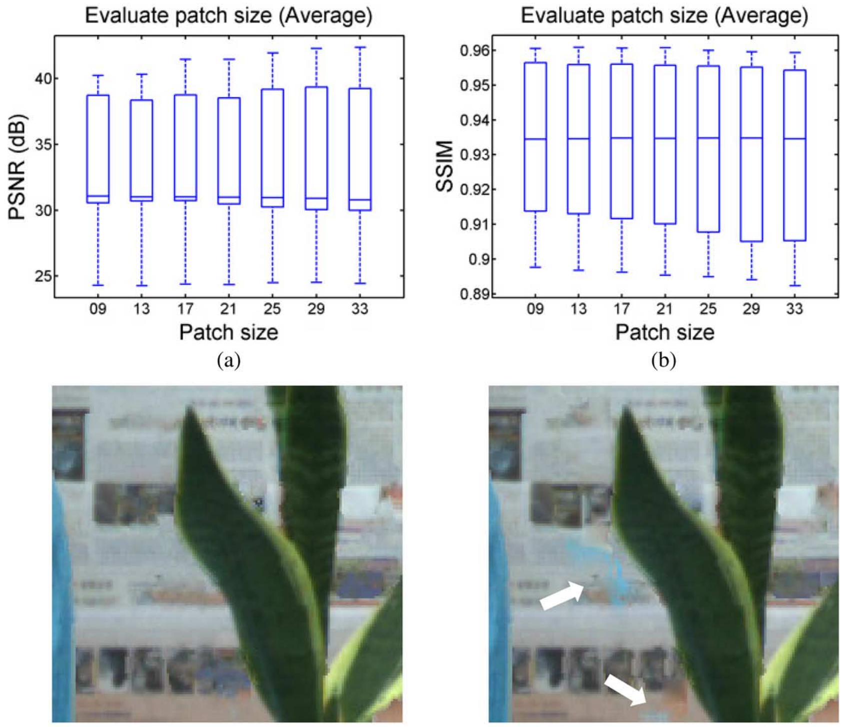
C. Assessment of the Texture Synthesis Algorithm

The most important texture synthesis parameters (cf. Sections V and VI) are:

- the search area A and its corresponding sub-sampling factor s;
- the patch size $L \times Q$;
- the weighting factor w_Ω (7)

Again, all tests were performed using the scenario with twice the regular baseline in order to have larger disoccluded areas. For reducing the complexity of estimating the texture synthesis parameters, a set of five key frames from each sequence and view (left and right) is used. The key frames were selected manually to ensure that all relevant scene contents as well as large disocclusions were considered. Furthermore, the DM was filled using the optimized k-means clustering settings determined in the previous section.

The search area A is an important parameter, which mainly depends on the content of the considered image. It has been observed that for the test sequences analyzed in this work, the view



(a) (b)

(c) (d)

Fig. 10. Influence of the patch size on the view synthesis accuracy. Overall values for (a) PSNR and (b) SSIM, over the whole test set. Subjective differences between the results after filling the disoccluded area with a patch of size (c) 9 × 9 and (d) 25 × 25.

synthesis performance is not very sensitive to the size of the search area. This may, however, be different for other sequences. It is possible to decrease the search complexity by sub-sampling A with a factor $s \in \mathbb{N}$. Increasing s decreases run time. However, there is depreciation in the quality of the results. It was found that it is adequate to set $A = 80 \times 80$ sample and $s = 2$, so that a reasonable compromise between complexity and sufficient quality is achieved.

Patches are assumed to be squares to simplify the evaluation, i.e., $L = Q$. Moreover, the initialization step is disabled, i.e., only the texture synthesis approach is taken into account. The influence of the patch size on the view synthesis results is shown in Fig. 10(a) and (b). No significant difference can be observed between the diverse patch size selections. Extensive viewing of synthesized views, however, show that a patch size of 9 × 9 $(L = Q = 9)$ yields better subjective results than larger patches, e.g., 25 × 25. The larger the block size is, the more likely it is for artifacts to occur. Fig. 10(c) and (d) illustrate the difference between the results obtained after the disoccluded area was filled with a patch size of 9 × 9 and 25 × 25. It can be seen that FG colors have been copied into the BG area with a patch size of 25 × 25.

The impact of the initialization of the uncovered regions in the textured images is depicted in Fig. 11. Note that the texture synthesis step after the initialization is performed with the optimized patch size $L = Q = 9$, $A = 80 \times 80$ and $s = 2$. It is seen that, for both measures [Fig. 11(a) and (b)], a quality improvement can be achieved by setting $w_\Omega \neq 0$ (7). When w_Ω is set to 0.2 instead of 0.0, the PSNR is increased by approximately 3 dB on average, while SSIM rises by 0.02. The gains are even larger for "Book Arrival" (S1), as shown in Fig. 11(c) and (d).

TABLE I
OPTIMIZED SETTINGS OF PROPOSED VIEW SYNTHESIS METHOD

Parameter	Value
A	80×80
s	2
M	32
N	32
L	9
Q	9
w_Ω	0.2

Increasing w_Ω further does, however, not yield further gains. Therefore, the final setting for this parameter is selected to be 0.2. Fig. 11(e)–(g) further illustrates the subjective results for the sequence "Book arrival" (S1).

D. Assessment of the Sprite Updating Algorithm

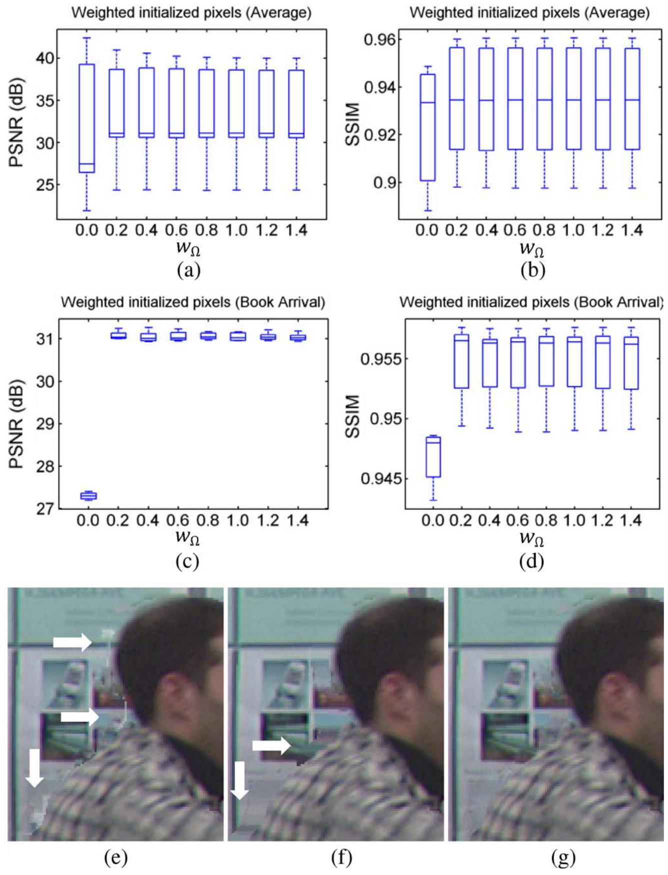
By using a BG sprite, temporal consistency is achieved in the virtual view. Nevertheless, the updating process highly depends on the quality of the DM (cf. Section IV). If unreliable DMs are used, inappropriate image information can be falsely copied into the sprite and propagate to subsequent frames. Therefore, the quality of the results from rendering can suffer. Vice versa, in the case of high quality DMs, the sprite updating works efficiently and leads to accurate temporally consistent synthesis results.

E. Overall System Evaluation

Given the experiments conducted in the previous sections, optimized parameter settings have been derived and summarized in Table I.

TABLE II
PSNR AND SSIM RESULTS BY THE PROPOSED FRAMEWORK, THE VIEW SYNTHESIS REFERENCE SOFTWARE, AND FEHN

Seq	Cam.	PSNR (dB)			SSIM		
		Prop.	MPEG	Fehn	Prop.	MPEG	Fehn
S1	8 → 9	**37.23**	36.06	35.73	**0.9837**	0.9828	0.9725
S1	10 → 9	**35.69**	35.15	34.66	**0.9827**	0.9810	0.9754
S1	8 → 10	**31.24**	30.25	29.80	**0.9552**	0.9525	0.9171
S1	10 → 8	**30.58**	30.30	29.46	**0.9549**	0.9524	0.9205
S2	6 → 7	47.29	**48.20**	47.63	**0.9272**	0.9267	0.9257
S2	6 → 8	40.37	**42.13**	40.84	**0.9301**	0.9284	0.9212
S2	8 → 6	**39.90**	38.54	37.85	**0.9444**	0.9425	0.9345
S2	8 → 7	**43.02**	41.54	41.41	**0.9532**	0.9524	0.9511
S3	4 → 6	25.30	**25.43**	24.69	0.8936	**0.8974**	0.8544
S3	4 → 5	30.84	**31.19**	30.64	0.9514	**0.9534**	0.9413
S3	6 → 4	**31.01**	30.37	28.98	0.9123	**0.9131**	0.8767
S3	6 → 5	**34.69**	34.67	33.89	0.9525	**0.9538**	0.9427
S4	5 → 3	**41.38**	34.98	34.15	**0.9840**	0.9800	0.9644
S4	5 → 4	**45.65**	37.64	36.86	**0.9868**	0.9849	0.9788
S4	5 → 6	**38.83**	37.85	36.98	**0.9867**	0.9846	0.9803
S4	5 → 7	**34.79**	33.37	32.47	**0.9799**	0.9756	0.9590



(a) (b)

(c) (d)

(e) (f) (g)

Fig. 11. Influence of the initialization step on the view synthesis accuracy. Overall values for (a) PSNR and (b) SSIM, for patch size of 9 × 9 over the whole test set. The objective results for the sequence "Book arrival" with (c) PSNR and (d) SSIM. Result after filling the disoccluded area (e) without initialization. (f) Result using the initialization step ($w_\Omega = 0.2$) without texture synthesis and (g) result using the initialization step ($w_\Omega = 0.2$) before texture synthesis.

The proposed approach is compared with the MPEG view synthesis reference software (VSRS, version 3.5) [32] with optimized parameters and to the disocclusion elimination method presented by Fehn [21]. In a rectified camera setup, the disocclusion elimination method fails to close holes on the left or right image border. Hence, the inpainting method introduced in [6] is used to fill these disocclusions. The achieved objective results, shown in Table II, are better than alternative approaches such as cropping and subsequently resizing the image.

In Table II the "camera" column (Cam.) represents the projection configuration, i.e., "8 → 9" refers to the synthesis of camera 9 from camera 8. Note that the PSNR and SSIM values (quality measures) correspond to the mean over all pictures of a sequence. The best result for each sequence is highlighted through bold face type. For the sequences "Book arrival" (S1), the proposed approach gives better SSIM and PSNR results than VSRS and Fehn [21]. For the "Lovebird1" (S2) sequence our algorithm shows better SSIM results. For the PSNR value, MPEG VSRS performs better for the case "6 → 8" and "6 → 7". Subjectively, in these cases the VSRS rendering is blurrier, while our results are sharper but noisier. For the sequence "Newspaper" (S3) VSRS achieves better overall results. As all our modules rely on the DM and the DM of "Newspaper" (S3) is particularly unreliable, visual and objective losses occur for the proposed algorithm. Nevertheless, some visual and objective gains can be obtained for the case "6 → 4" and "6 → 5" (see Table II and Fig. 4(d)–(f); electronic magnification may be required). For the sequences "Mobile" (S4) characterized by a highly structured BG, the proposed approach gives better SSIM and PSNR results than VSRS and Fehn [21].

In addition to the objective measurements, Figs. 13 and 14 show some subjective results. In Figs. 13 and 14(a), the original reference pictures 51 and 184 of the "Book arrival" (S1) and "Mobile" (S4) sequences are shown. In Fig. 14(b), the rendered images are shown [baseline extension, warping camera 8 to the position of camera 10 (S1), warping camera 5 to the position of camera 3 (S4)]. The disoccluded areas are marked white (S1) or black (S4). In Fig. 14 (c) and (d), the final BG sprite and its associated DM are shown. The final rendering results by the proposed approach are shown in Fig. 14(e). The result by VSRS (S1) and Fehn [21] (S4) are shown in Fig. 14(f). Note that the proposed approach yields sharper edges than VSRS (Fig. 13)

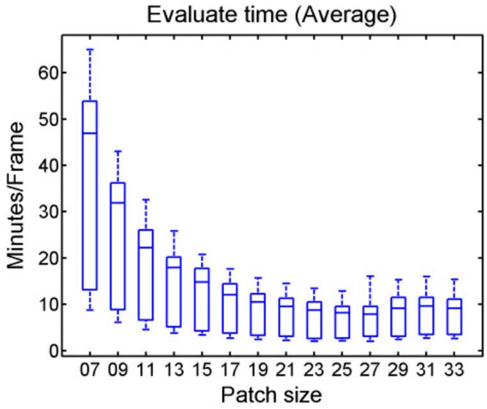


Fig. 12. Evaluation of the run-time using different patch sizes.

and FG objects maintain their shape (Fig. 14). Fig. 14(g) and (h) shows magnifications of the squared areas in Fig. 14(e) and (f), where the results for the proposed algorithm are shown on the left side and the results for VSRS on the right. FG information is correctly omitted by our proposed filling method [Fig. 13(g)], which yields significantly improved synthesis results compared to VSRS. As can be seen in Fig. 13(h) on the poster in the BG, details are again well preserved by our method. In Fig. 14(h) it can be seen that the algorithm proposed by Fehn [21] leads to annoying distortions in the BG area and the FG objects. The proposed method preserves the shape of the objects [Fig. 14(g)]. The BG in "Mobile" (S4) is highly structured but even these disocclusions can be reconstructed satisfactorily with the proposed method. Fig. 14(i) and (j) shows the objective results. Gains are achieved in PSNR and SSIM.

F. Complexity Assessment

The complexity of the proposed algorithm is mainly dominated by the following three aspects:

1) the search area A with the corresponding subsampling factor s, of the texture synthesis approach;
2) the patch size used in the texture refinement step;
3) the utilized cloning method used in the updating process.

The other functions are less time consuming and their contribution to the overall complexity is rather small.

A PC with an Intel Xeon CPU and 4-GB RAM was used in our experiments. Our software is currently implemented in MATLAB. The optimized settings given in Table I were used.

According to the results obtained, varying the search area A and the subsampling factor s, strongly influences the complexity of the proposed approach. The complexity increases by a factor of approximately 1.5 when A is doubled. On the other hand, when s is increased from 1 to 2, the complexity reduces by a factor of approximately 1.31. Increasing s from 1 to 4 yields a complexity reduction of approximately 3.24.

To update the actual frame from the BG sprite, the covariant cloning is used to fit the BG data into the frame (see Section VII-B). For every sample position to be updated from the BG, a linear equation has to be solved. Hence, the complexity is proportional to the number of samples which are copied from the BG sprite to the actual frame, which corresponds to linear growth of complexity.

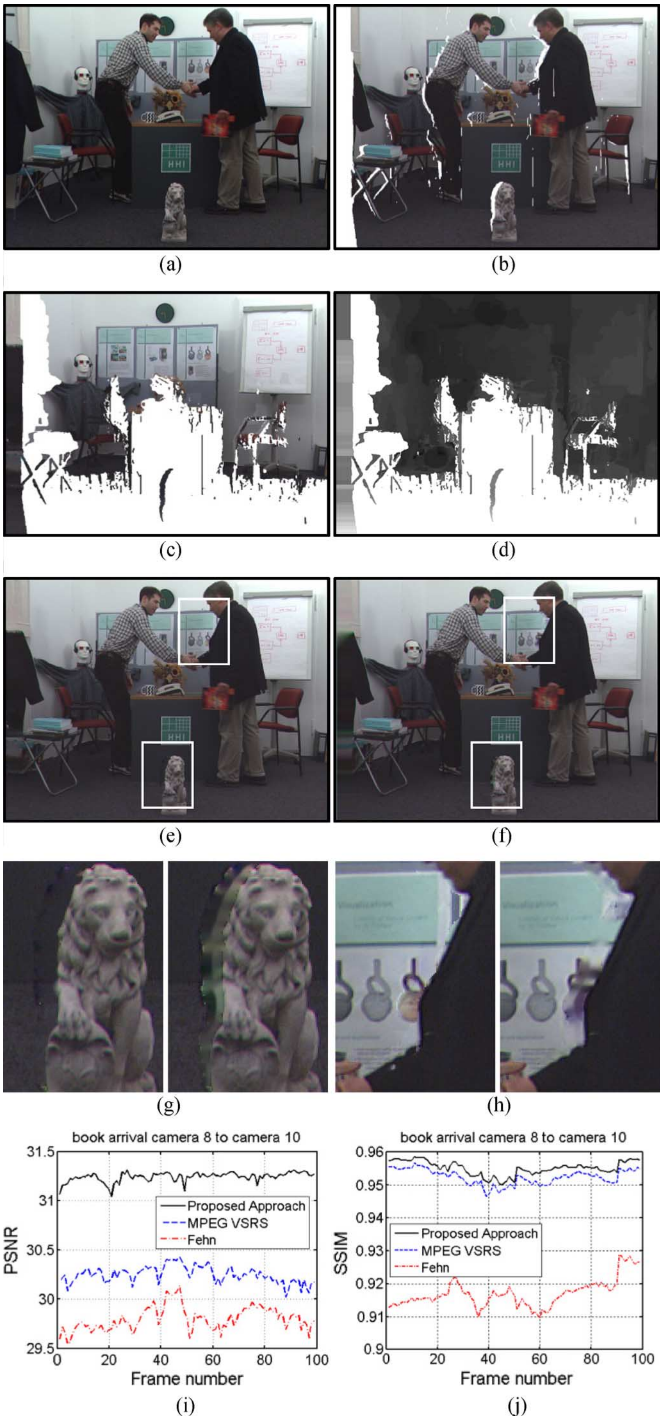


Fig. 13. DIBR results for the "Book arrival" sequence. (a) Original reference image 51. (b) Rendered image with disoccluded area marked white. (c) Final BG sprite with unknown area marked white and its associated DM (d). (e) Result of picture 51 by the proposed approach. (f) Result of VSRS for the same picture. (g), (h) Magnified results. Left: proposed approach. Right: VSRS. (i), (j) Objective results for all frames of the sequence.

Fig. 12 depicts the results obtained by varying the (square) patch size. Furthermore, the results are generated with the key frames used in Section VII-C, i.e., the run time represents the mean value of the different single images (no time consistency is available) evaluated with the same patch size. It can be seen that the complexity is approximately inversely proportional to the patch size growth. This relates to the fact that larger patches

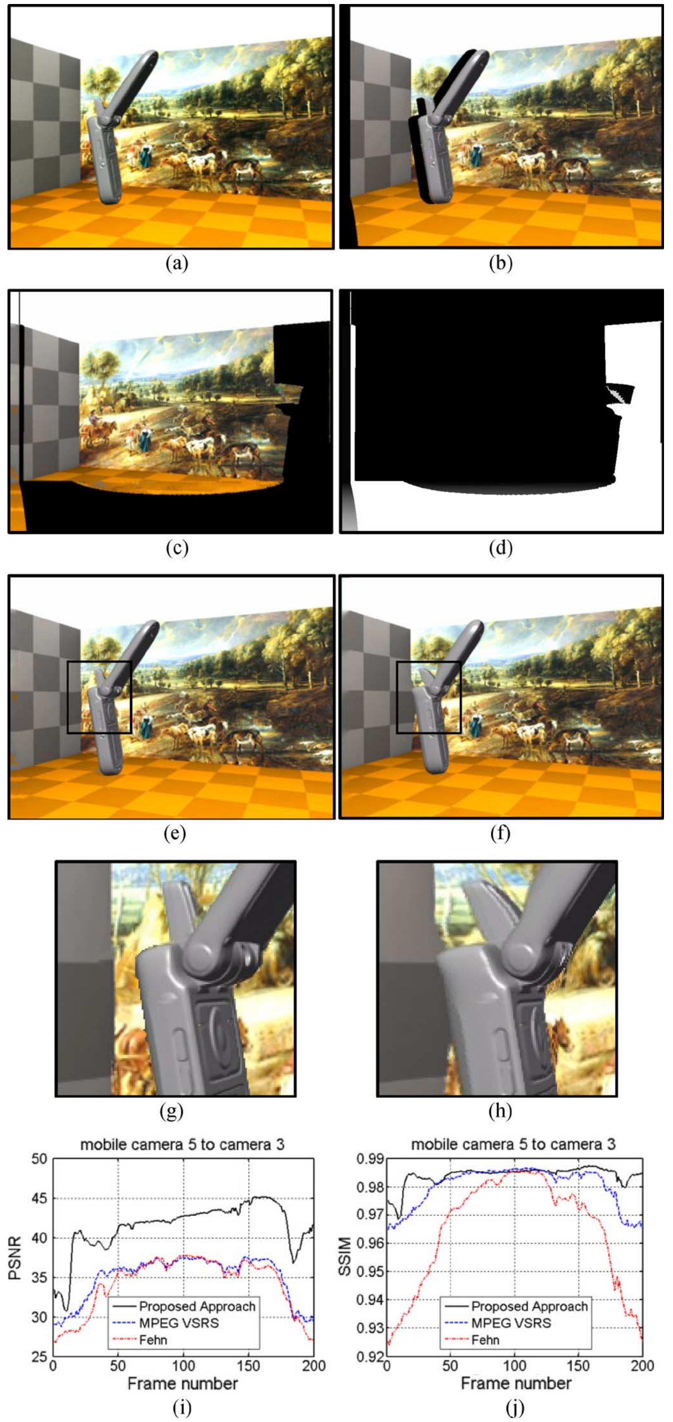


Fig. 14. DIBR results for the "Mobile" sequence. (a) Original reference image 184. (b) Rendered image with disoccluded area marked black. (c) Final BG sprite with unknown area marked black and (d) its associated DM (unknown area marked white). (e) Result of picture 184 by the proposed approach. (f) Result of Fehn [21] for the same picture. (g), (h) Magnified results. (g) Proposed approach. (h) Fehn [21]. (i), (j) Objective results for all frames of the sequence.

cover more unknown pixels. Hence, fewer search iterations have to be run. If texture synthesis with time consistency (using sprite update) is applied with a patch of size 9 × 9, the run time decreases by a factor of ∼3.2 compared to texture synthesis without time consistency (cf. Fig. 12). However, in applications where complexity is of more importance than quality, a patch size of 25 × 25 appears to be the better choice.

Note that the figure depicted here is rather useful for a relative than for an absolute evaluation, as a hardware/C++ software implementation of the same approach will have different absolute complexity characteristics.

VIII. Conclusion and Future Work

We have described a new hole-filling approach with inpainting methods for DIBR. The algorithm works for large baseline extensions and generates spatio-temporally consistent rendering results. Each virtual view image featuring disocclusions is compensated using image information from a causal picture neighborhood via a BG sprite. Residual uncovered areas are initially coarsely estimated and then refined using texture synthesis. We have shown that the presented approach yields subjective and objective gains compared to state of the art view synthesis, given reasonable DM quality. However, depth estimation inconsistencies especially at foreground–background transitions may lead to considerable degradation of the rendering results. This dependency will therefore be reduced in future work. Our examinations also showed that the lack of an adequate perceptual measure for 3-D content, hampers a fully optimized configuration of our view synthesis algorithm. Towards that end, extensive subjective experiments will be carried out in future work. Additionally, the problem of global and local BG motion as well as complexity issues will be addressed.

Acknowledgment

The authors would like to thank M. Müller and P. Kauff for bringing our attention to the usefulness of advanced synthesis methods in 3-D video. The authors would also like to thank the Gwangju Institute of Science and Technology, Korea, the Electronic and Telecommunications Research Institute/MPEG Korea Forum, and Philips for providing the "Newspaper," "Lovebird1," and "Mobile" sequences, respectively.

References

[1] O. Schreer, P. Kauff, and T. Sikora*, 3D Video Communication: Algorithms, Concepts and Real-Time Systems in Human Centred Communication*. New York: Wiley, 2005.

[2] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based inpainting for view synthesis in free viewpoint television and 3-D video," in *Proc. Picture Coding Symp.*, Chicago, IL, May 2009, pp. 233–236.

[3] C. L. Zitnik, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2004, pp. 600–608.

[4] S. Zinger, L. Do, and P. H. N. de With, "Free-viewpoint depth image based rendering," *J. Vis. Commun. Image Representation*, vol. 21, no. 5–6, pp. 533–541, 2010.

[5] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3-D warping using depth information for FTV," *IEEE J. Signal Process.*, vol. 24, no. 1–2, pp. 65–72, Jan.–Feb. 2009.

[6] A. Telea, "An image inpainting technique based on the fast marching method," *Int. J. Graphic Tools*, vol. 9, no. 1, pp. 25–36, 2004.

[7] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, New Orleans, LA, Jul. 2000, pp. 417–424.

[8] K. Müller, A. Smolic, K. Dix, P. Merkle, P. Kauf, and T. Wiegand, "View synthesis for advanced 3-D video systems," *EURASIP J. Image Video Process.*, vol. 2008, 2008, Art. ID 438148.

[9] W. R. Mark, "Post-rendering 3-D image warping: Visibility, reconsruction, and performance for depth-image warping," Ph.D. dissertation, Graph. Image Process. Lab., Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, 1999.

[10] X. Jiufei, X. Ming, L. Dongxiao, and Z. Ming, "A new virtual view rendering method based on depth image," in *Proc. Asia–Pacific Conf. Wearable Computing Syst.*, Shenzhen, China, Apr. 2010, pp. 147–150.

[11] W.-Y. Chen, Y.-L. Chang, S.-F. Lin, L.-F. Ding, and L.-G. Chen, "Efficient depth image based rendering with edge dependent depth filter and interpolation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 1314–1317.

[12] L. Zhang and W. J. Tamm, "Stereoscopic image generation based on depth images for 3-D TV," *IEEE Trans. Broadcasting*, vol. 51, no. 2, pp. 191–199, Jun. 2005.

[13] P.-J. Lee and Effendi, "Adaptive edge-oriented depth image smoothing approach for depth image based rendering," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcasting*, Shanghai, China, Mar. 2010, pp. 1–5.

[14] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Tampa, FL, Dec. 2008, pp. 1–4.

[15] M. Schmeing and X. Jiang, "Depth image based rendering: A faithful approach for the disocclusion problem," in *Proc. 3DTV-Conf.: True Vision—Capture, Transmission and Display of 3-D Video*, Tampere, Finland, Jun. 2010, pp. 1–4.

[16] K.-Y. Chen, P.-K. Tsung, P.-C. Lin, H.-J. Yang, and L.-G. Chen, "Hybrid motion/depth-oriented inpainting for virtual view synthesis in multiview applications," in *Proc. 3DTV-Conf.: True Vision—Capture, Transmission and Display of 3-D Video*, Tampere, Finland, Jun. 2010, pp. 1–4.

[17] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG4-AVC), v1, May 2003; v2, Jan. 2004; v3 (with FRExt), Sep. 2004; v4, Jul. 2005.

[18] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[20] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, "Improved novel view synthesis from depth image with large baseline," in *Proc. Int. Conf. Pattern Recognit.*, Tampa, FL, Dec. 2008, pp. 1–4.

[21] C. Fehn, "Depth Image Based Rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Proc. SPIE Stereoscopic Disp. Virtual Reality Syst. XI*, San Jose, CA, Jan. 2004, pp. 93–104.

[22] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[23] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1995, pp. 229–238.

[24] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–71, 2000.

[25] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vis.*, pp. 91–109, Feb. 2004.

[26] J. Hayes and A. Efros, "Scene completion using millions of photographs," in *Proc. ACM SIGGRAPH*, San Diego, CA, Aug. 2007, pp. 1–7.

[27] J. S. DeBonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proc. ACM SIGGRAPH*, 1997, pp. 361–368.

[28] M. Ashikhmin, "Synthesizing natural textures," in *Proc. ACM Symp. Interactive 3-D Graphics*, New York, 2001, pp. 217–226.

[29] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," in *Proc. ACM SIGGRAPH*, San Diego, CA, Jul. 2003, pp. 277–286.

[30] P. Ndjiki-Nya, M. Köppel, D. Doshkov, and T. Wiegand, "Automatic structure-aware inpainting for complex image content," in *Proc. Int. Symp. Visual Computing*, Las Vegas, NV, Dec. 2009, pp. 1144–1156.

[31] M. Tanimoto, T. Fujii, and K. Suzuki, "Depth Estimation Reference Software (DERS) 5.0," Lausanne, Switzerland, ISO/IEC JTC1/SC29/WG11 M16923, Oct. 2009.

[32] M. Tanimoto, T. Fujii, and K. Suzuki, "View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS 2.0)," Lausanne, Switzerland, ISO/IEC JTC1/SC29/WG11 M16090, Feb. 2008.

[33] C. M. Bishop*, Neural Networks for Pattern Recognition*. Oxford: Oxford Univ. Press, 1995.

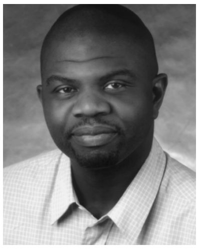
[34] T. G. Georgiev, "Photoshop healing brush: A tool for seamless cloning," in *Proc. Eur. Conf. Comput. Vis.*, Prague, Czech Republic, May 2004, pp. 1–8.

[35] T. G. Georgiev, "Covariant derivates and vision," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, 2006, pp. 56–69.

[36] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH*, San Diego, CA, Jul. 2003, pp. 313–318.

[37] H. Lakshman, M. Köppel, P. Ndjiki-Nya, and T. Wiegand, "Image recovery using sparse reconstruction based texture refinement," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Dallas, TX, Mar. 2010, pp. 786–789.

[38] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2005, pp. 861–868.



Patrick Ndjiki-Nya (M'98) received the Dipl.-Ing. title (corr. to M.S degree) and the Ph.D. degree from the Technical University of Berlin, Berlin, Germany, in 1997 and 2008, respectively.

He has developed an efficient method for content-based video coding, which combines signal theory with computer graphics and vision. His approaches are currently being evaluated in equal or similar form by various companies and research institutions in Europe and beyond. From 1997 to 1998, he was significantly involved with the development of a flight simulation software at Daimler-Benz AG. From 1998 to 2001, he was a Development Engineer with DSPecialists GmbH, where he was concerned with the implementation of algorithms for digital signal processors (DSPs). During the same period, he researched content-based image and video features with the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, with the purpose of implementation in DSP solutions from DSPescialists GmbH. Since 2001, he has been with the Fraunhofer Heinrich Hertz Institute solely, where he was Project Manager initially and Senior Project Manager from 2004 on. He has been Group Manager since 2010.

Martin Köppel received the Dipl.-Ing. degree in media technologies from the Technical University of Ilmenau, Ilmenau, Germany, in 2008.

Before then, he was a Student Engineer with the Institut of Microelectronic- and Mechatronic Systems gGmbH, Ilmenau, Germany, from 2004 to 2006. He was a Teaching Assistant with the Technical University of Ilmenau from 2006 to 2007. He joined the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, in 2007, and has been working there as a Research Associate since 2008. His research interests are in the fields of image and video processing. He has been involved in several projects in the areas of texture synthesis, view synthesis, video coding, and 3-D video.

Dimitar Doshkov received the Dipl.-Ing. degree in telecommunication engineering from the University of Applied Sciences of Berlin, Berlin, Germany, in 2008.

He was with miControl Parma & Woijcik OHG from 2004 to 2005. In 2006, he joined Samsung SDI Germany GmbH as a Trainee. He has been with the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, since 2007, where he has been a Research Associate since 2008. His research interests include image and video processing, as well as computer vision and graphics. He has been involved in several projects focused on image and video synthesis, video coding, and 3-D video.

Haricharan Lakshman received the B.E. degree from the National Institute of Technology Karnataka, Karnataka, India, in 2002, and the M.S. degree in electrical engineering from the University of Erlangen-Nuremberg, Erlangen, Germany, in 2008.

From 2002 to 2005, he was an Engineer with Ittiam Systems, India. From 2005 to 2006, he was with the Audio Group of the Fraunhofer-Institute for Integrated Circuits, Erlangen, Germany. In 2008, he joined the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, as a Research Associate. His research interests include image and video coding and postprocessing.

Philipp Merkle (S'06) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Berlin, Berlin, Germany, in 2006.

He joined the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, in 2003, where he has been as a Research Associate since 2006. He has been involved in several projects focused on multiview video coding, 3-D television, free viewpoint video, and 3-D scene reconstruction. His research interests include 3-D video, representation and compression of multiview-video-plus-depth scenes, free viewpoint video, and 2-D and 3-D video-based rendering. He has been involved in ISO standardization activities where he contributed to the development of the MPEG-4 multiview video coding standard.

Mr. Merkle was the recipient of the "Rudolf-Urtel Award" of the German Society for Technology in TV and Cinema (FKTG) for his work on multiview video coding in 2006.

Karsten Müller (M'98–SM'07) received the Dipl.-Ing. degree and the Dr.-Ing. degree in electrical engineering from the Technical University of Berlin, Berlin, Germany, in 1997 and 2006, respectively.

He has been with the Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin, Germany, since 1997, where he is currently Project Manager for 3-D video projects. His research interests are mainly in the field of representation, coding, and reconstruction of 3-D scenes in free viewpoint video scenarios and coding, multiview applications and combined 2-D/3-D similarity analysis. He has been involved with MPEG standardization activities in 3-D video coding and content description.

Thomas Wiegand (M'05–SM'08–F'11) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Hamburg, Germany, in 1995, and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Erlangen, Germany, in 2000.

He is a Professor with the Department of Electrical Engineering and Computer Science, Berlin Institute of Technology, Berlin, Germany, chairing the Image Communication Laboratory, and is jointly heading the Image Processing Department, Fraunhofer Institute for Telecommunications—Heinrich Hertz Institute, Berlin. He joined the Heinrich Hertz Institute in 2000 as the head of the Image Communication Group in the Image Processing Department. His research interests include video processing and coding, multimedia transmission, as well as computer vision and graphics. From 1993 to 1994, he was a Visiting Researcher with Kobe University, Japan. In 1995, he was a Visiting Scholar with the University of California at Santa Barbara. From 1997 to 1998, he was a Visiting Researcher with Stanford University and served as a consultant to 8 × 8, Inc., Santa Clara, CA. From 2006 to 2008, he was a Consultant to Stream Processors, Inc., Sunnyvale, CA. From 2007 to 2009, he was a Consultant to Skyfire, Inc., Mountain View, CA. Since 2006, he has been a Member of the Technical Advisory Board of Vidyo, Inc., Hackensack, NJ. Since 1995, he has been an active participant in standardization for multimedia with successful submissions to ITU-T VCEG, ISO/IEC MPEG, 3GPP, DVB, and IETF. In October 2000, he was appointed as the Associated Rapporteur of ITU-T VCEG. In December 2001, he was appointed as the Associated Rapporteur/Co-Chair of the JVT. In February 2002, he was appointed as the Editor of the H.264/MPEG-4 AVC video coding standard and its extensions (FRExt and SVC). From 2005–2009, he was Co-Chair of MPEG Video.

Prof. Wiegand was the recipient of the SPIE VCIP Best Student Paper Award in 1998 and the Fraunhofer Award and the ITG Award of the German Society for Information Technology in 2004. The projects that he co-chaired for development of the H.264/MPEG-4 AVC standard have been recognized by the 2008 ATAS Primetime Emmy Engineering Award and a pair of NATAS Technology & Engineering Emmy Awards. In 2009, he was the recipient of the Innovations Award of the Vodafone Foundation, the EURASIP Group Technical Achievement Award, and the Best Paper Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. In 2010, he was the recipient of the Eduard Rhein Technology Award. He was a guest editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY for its Special Issue on the H.264/AVC Video Coding Standard in July 2003, its Special Issue on Scalable Video Coding—Standardization and Beyond in September 2007, and its Special Section on the Call for Proposal on High-Efficiency Video Coding in December 2010. Since January 2006, he has been an associate editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.