Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 438148, 11 pages
doi:10.1155/2008/438148

Research Article

View Synthesis for Advanced 3D Video Systems

Karsten Müller, Aljoscha Smolic, Kristina Dix, Philipp Merkle, Peter Kauff, and Thomas Wiegand

Image Processing Department, Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, Einsteinufer 37, 10587 Berlin, Germany

Correspondence should be addressed to Karsten Müller, kmueller@hhi.de

Received 31 March 2008; Accepted 20 November 2008

Recommended by Stefano Tubaro

Interest in 3D video applications and systems is growing rapidly and technology is maturating. It is expected that multiview autostereoscopic displays will play an important role in home user environments, since they support multiuser 3D sensation and motion parallax impression. The tremendous data rate cannot be handled efficiently by representation and coding formats such as MVC or MPEG-C Part 3. Multiview video plus depth (MVD) is a new format that efficiently supports such advanced 3DV systems, but this requires high-quality intermediate view synthesis. For this, a new approach is presented that separates unreliable image regions along depth discontinuities from reliable image regions, which are treated separately and fused to the final interpolated view. In contrast to previous layered approaches, our algorithm uses two boundary layers and one reliable layer, performs image-based 3D warping only, and was generically implemented, that is, does not necessarily rely on 3D graphics support. Furthermore, different hole-filling and filtering methods are added to provide high-quality intermediate views. As a result, high-quality intermediate views for an existing 9-view auto-stereoscopic display as well as other stereo- and multiscopic displays are presented, which prove the suitability of our approach for advanced 3DV systems.

Copyright © 2008 Karsten Müller et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

3D video (3DV) provides the viewer with a depth perception of the observed scenery. This is also referred to as stereo, which is, however, a term too restricted to the classical technology of using 2 videos. Recently, 3DV gains rapidly increasing attention spanning systems and applications from mobile phones to 3D cinema [1]. Technology is maturating covering the whole processing chain from camera systems to 3D displays. Awareness and interest are growing on consumer side, who wish to experience the extended visual sensation, as well as on business side including content providers, equipment producers, and distributors.

Creating a 3D depth impression requires that a viewer looking at a 3D display sees a different view with each eye. These views must correspond to images taken from different viewpoints with human eye distance. A 3D display emits two or more views at the same time and ensures that a viewer always sees such a stereo pair from a certain viewpoint [2]. Specific glasses based on anaglyph, polarization, or shutter technology were necessary to achieve this in the past but are today still appropriate for a wide range of applications. For instance, 3D cinema applications based on glasses (such as IMAX theatres) are well established. In a cinema theatre, the user is sitting in a chair without much possibility to move and is usually paying almost full attention to the presented movie. Wearing glasses is widely accepted in such a scenario and motion parallax is not a big issue. 3D cinema with display technology based on glasses is therefore expected to remain the standard over the next years. This market is expected to grow further and more and more movies are produced in 2D for classical cinema as well as in a 3D version for 3D-enabled theatres. It is expected that this will broaden awareness of users and with this also increase the acceptance and create demand for 3DV applications in the home.

In a living room environment, however, the user expectations are very different. The necessity to wear glasses is considered as a main obstacle for success of 3D video in home user environments. Now, this is overcome with multiview autostereoscopic displays [2]. Several images are emitted at the same time but the technology ensures that users only see a stereo pair from a specific viewpoint. 3D

displays are on the market today that are capable of showing 9 or more different images at the same time, of which only a stereo pair is visible from a specific viewpoint. With this, multiuser 3D sensation without glasses is enabled, for instance, in a living room. A group of people may enjoy a 3D movie in the familiar sofa-TV environment without glasses but with all social interactions that we are used to. When moving around, a natural motion parallax impression can be supported if consecutive views are arranged properly as stereo pairs.

However, transmitting 9 or more views of the same 3D scenery from slightly different viewpoints to the home user is extremely inefficient. The transmission costs would not justify the additional value. Fortunately, alternative 3D video formats allow for reducing the raw data rate significantly. When using the multiview video plus depth (MVD) format only a subset M of the N display views is transmitted. For those M video streams, additional per-pixel depth data is transmitted as supplementary information. At the receiver depth-image-based rendering (DIBR) is applied to interpolate all N display views from the transmitted MVD data [3]. The advanced 3DV system concept based on MVD and DIBR is presented in more detail in Section 2.

The success of this concept relies on the availability of high-quality intermediate view synthesis algorithms. A general formulation of such DIBR or 3D warping is given in Section 3. DIBR is known to produce noticeable artifacts that especially occur along object boundaries with depth discontinuities. Section 4, therefore, introduces a novel DIBR algorithm, where depth discontinuities are treated in a layered approach with image regions marked as reliable and unreliable areas. Results and improvements over standard 3D warping are presented in Section 5. Finally, Section 6 concludes the paper.

2. 3D VIDEO SYSTEM CONCEPT

Multiview autostereoscopic displays support head motion parallax viewing and multiuser applications without the necessity to wear glasses. They are in the center of the advanced concept for 3D video home systems considered in this paper. High-resolution LCD screens with slanted lenticular lens technology as commercially available, for instance, from Philips [4] are capable of displaying 9 and more simultaneous views. The principle is illustrated in Figure 1. At position 1, a user sees views 1 and 2 with right and left eyes, respectively, only. At another position 3, a user sees views 6 and 7, hence multi-user 3D viewing is supported.

Head motion parallax viewing can be supported as follows. If a user in Figure 1 moves from position 1 to position 2, views 2 and 3 are visible with the right and left eyes, respectively. If all views are properly arranged, that is, views 1 and 2, then views 2 and 3, and so on are stereo pairs with proper human eye distance baseline, then a user moving in front of such a 3D display system will perceive a 3D impression with head motion parallax. Disocclusions and occlusions of objects in the scenery will be perceived depending on their depth in the 3D scene. However, this effect will not be seamless but restricted to a number of predefined positions equal to $N - 1$ stereo pairs.

Thus, multiview autostereoscopic displays process N synchronized video signals showing the same 3D scene from slightly different viewpoints. Compared to normal 2D video, this is a tremendous increase of raw data rate. It has been shown that specific multiview video coding (MVC) including inter-view prediction of video signals taken from neighboring viewpoints can reduce the overall bit rate by 20% [5], compared to independent coding of all video signals (simulcast). This means a reduction by 20% of the single video bit rate multiplied by N. For a 9-view display, MVC, therefore, still requires 7.2 times the corresponding single video bit rate. Such an increase is clearly prohibitive for the success of 3DV applications. Further, it has also been shown in [5] that the total bit rate of MVC increases linearly with N. Future displays with more views would, therefore, require even higher total bit rates. Finally, fixing the number of views in the transmission format as done with MVC does not provide sufficient flexibility to support any type of current and future 3D displays.

For 2-view displays (or small number of views displays), a different approach was demonstrated to provide both high compression efficiency as well as extended functionality. Instead of transmitting a stereo video pair, one video and an associated per-pixel depth map is used. The depth map assigns a scene depth value to each of the pixels of the video signal, and with that provides a 3D scene description. The depth map can be treated as monochromatic video signal and coded using available video codecs. This way video plus depth (V + D) is defined as 3DV data format [6]. A corresponding standard known as MPEG-C Part 3 has been recently released by MPEG [7, 8]. From decoded V + D, a receiver can generate a second video as stereo pair by DIBR. Experiments have shown that depth data can be compressed very efficiently in most cases. Only around 10–20% of the bit rate necessary for the corresponding color video are required to compress depth at a sufficient quality. This means that the final stereo pair rendered using this decoded depth is of same visual quality as if the 2 video signals were transmitted instead. However, it is known that DIBR introduces artifacts. Generating virtual views requires extrapolation of image content to some extent. From a virtual viewpoint, parts of the 3D scene may become visible that are occluded behind foreground objects in the available original video. If the virtual viewpoint is close to the original camera position (e.g., corresponding to V1 and V2 in Figure 1) masking of uncovered image regions works well with limited artifacts. Therefore, V + D is an excellent concept for 3D displays with a small number of views. However, with increasing distance of the virtual viewpoint also the extrapolation artifacts increase. The concept of V + D is, therefore, not suitable for 3DV systems with a large number of views and motion parallax support over a wide range.

In consequence, neither MVC nor V + D are useful for advanced 3D display systems with a large number of views. The solution presented here is the extension and combination to MVD as illustrated in Figure 1. 9 views V1–V9 are displayed. Direct encoding with MVC would be

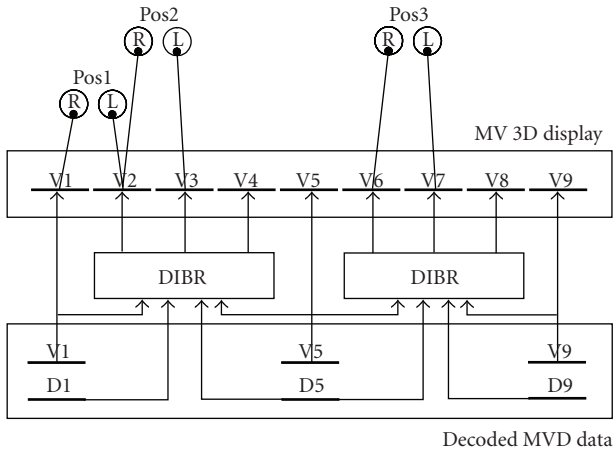

Figure 1: Advanced 3DTV concept based on MVD; Pos: viewpoint, R: right eye, L: left eye, V: view/image, D: depth.

highly inefficient. Transmitting only one video with a depth map, for example, V5 + D5 would result in unacceptable quality of outer views. Using the MVD format, a subset of $M = 3$ views with depth maps is transmitted to the receiver. Intermediate views V2–V4 and V6–V8 are generated by DIBR. They are close enough to available original views to minimize extrapolation errors. Further, they can be interpolated from 2 directions (left and right neighbor views), thus the problem of uncovering can be widely minimized. For instance, regions to be generated for the virtual view that are occluded in the left view are very likely visible in the right view. However, there is still the possibility that parts are occluded in both original views and finally have to be extrapolated.

This advanced 3DV system concept includes a number of sophisticated processing steps that are partially unresolved and still require research. Acquisition systems still have to be developed and optimized, which includes multicamera systems, possibly depth capture devices, as well as other types of maybe only supporting sensors and sources of information such as structured light [9, 10]. Sender side signal processing includes a lot of advanced algorithms such as camera calibration, color correction, rectification, segmentation as well as depth estimation or generation. The latter is crucial for DIBR since any error of depth estimation results in reduced quality of rendered output views. It is a topic widely studied in computer vision literature, which may include semiautomatic processing as well [11–14]. Especially in depth estimation, the resulting depth maps exhibit errors at object boundaries with different depths. Usually, depth edges are smoothed such that a foreground-background-separation cannot be applied properly. In such cases, depth enhancement is required for high-quality rendering, for example, depth edge amplification by high-pass filtering with additional color and depth edge alignment. In our approach, we only consider high-quality depth maps as input and part of an MVD data format. Optimum parameterization of the generic 3DV format still needs to be investigated, including the number of transmitted views with depth and the setting/spacing. Most efficient compression of the MVD data is still to be found, especially optimum treatment of depth. As usual, transmission issues have to be considered for different channels. Finally, after decoding, the N output views have to be rendered out of the decoded MVD data. Here, high quality with few artifacts is crucial for the success of the whole concept. The rest of this paper presents an efficient solution for high-quality rendering at receiver side.

3. GENERAL FORMULATION OF DEPTH-BASED INTERMEDIATE VIEW SYNTHESIS

Within the 3DV framework, we assume a given input data in the form of color data l_k, depth data d_k and camera parameters for each original view k. This data may be provided by a capturing process for l_k and an associated depth camera or depth estimation process for d_k. For the latter, depth map improvement may be required as described above to provide sharply defined depth edges, required by our layered approach. As an example, the original views for the advanced 3DTV concept are shown in Figure 1 bottom for $k \in \{1, 5, 9\}$. Camera parameters for each original view k are given in the form of intrinsic parameters (focal length, sensor scaling, and principle point) in the intrinsic matrix $\mathbf{K}_k$ and extrinsic parameters (rotation, translation) in the extrinsic matrix $[\mathbf{R}_k|\mathbf{t}_k]$ with rotation matrix $\mathbf{R}_k$ and translation vector $\mathbf{t}_k$. They can be obtained by classical camera calibration algorithms [15–17]. Usually, extrinsic and intrinsic matrix are multiplied to obtain the projection matrix $\mathbf{P}_k = \mathbf{K}_k[\mathbf{R}_k|\mathbf{t}_k]$ which projects 3D world points into the image plane of original camera view k. Thus, an original view is given by

$$l_k(u_k, v_k), \qquad d_k(u_k, v_k), \qquad \mathbf{P}_k, \tag{1}$$

at each pixel position (u_k, v_k).

The given framework provides a number of sparse original cameras, in the form of (1). The task of view synthesis is to provide dense intermediate views between any pair of adjacent original cameras. For the mathematic derivation of this interpolation process, two original views k and n are given according to (1). For an arbitrary virtual view position between the two cameras, an interpolation parameter $\lambda \in [0 \cdots 1]$ is introduced, where $\lambda = 0$ refers to the first original viewing position, $\lambda = 1$ to the second, and $\lambda = 0.5$, for instance, defines the middle position. For the intermediate view $l_\lambda(u_\lambda, v_\lambda)$, the associated intrinsic and extrinsic matrices are calculated first as follows:

$$\begin{aligned} \mathbf{K}_\lambda &= (1-\lambda)\mathbf{K}_k + \lambda\mathbf{K}_n, \\ \mathbf{t}_\lambda &= (1-\lambda)\mathbf{t}_k + \lambda\mathbf{t}_n, \\ \mathbf{R}_\lambda &= \mathrm{slerp}(\mathbf{R}_k, \mathbf{R}_n, \lambda). \end{aligned} \tag{2}$$

Here, all parameters are linearly interpolated, except the parameters in the rotation matrix, where spherical linear interpolation [18] is used to preserve the matrix orthonormality. For this, the column vectors of both matrices $\mathbf{R}_k$ and $\mathbf{R}_n$ are interpolated separately to obtain the column vectors of

$\mathbf{R}_\lambda$. This calculation is shown exemplary for the first column vector $\mathbf{R}_\lambda(i,1)$ of matrix $\mathbf{R}_\lambda$:

$$\begin{aligned}\mathbf{R}_\lambda(i,1) &= \operatorname{slerp}(\mathbf{R}_k(i,1),\mathbf{R}_n(i,1),\lambda)\\ &= \frac{\sin\left((1-\lambda)\alpha_i\right)\mathbf{R}_k(i,1)+\sin\left(\lambda\alpha_i\right)\mathbf{R}_n(i,1)}{\sin\left(\alpha_i\right)},\\ &\text{with } \alpha_i = \arccos\left(\mathbf{R}_k(i,1)\cdot\mathbf{R}_n(i,1)\right).\end{aligned} \tag{3}$$

For $\alpha_i \to 0$, the associated column vectors are in parallel and the spherical linear interpolation simplifies to an ordinary linear interpolation. The other two column vectors are calculated accordingly. From the interpolated intrinsic and extrinsic matrices, the intermediate view projection matrix is calculated accordingly as follows: $\mathbf{P}_\lambda = \mathbf{K}_\lambda[\mathbf{R}_\lambda|\mathbf{t}_\lambda]$. Other methods calculate intermediate view projections from three independent original views based on tensor spaces [19] and disparity scaling [20–23] to address pixel positions in intermediate views. For the interpolation, all color values from both original camera views $l_k(u_k,v_k)$ and $l_n(u_n,v_n)$ are projected into the intermediate view by projecting their associated pixel positions.

The following considerations are carried out for view k only, since the calculations are similar for view n: For view k, the associated pixel position (u_k,v_k) is projected into 3D space first, using the inverse projection matrix $\mathbf{P}_k^{-1}$. This projection is ambiguous, since a single 2D pixel point from the camera plane is projected onto the straight line through the camera focal point and pixel position point. Therefore, the depth data $d_k(u_k,v_k)$ is required to determine the exact 3D position. Often, depth data is provided in scaled and quantized form, such that the true values $z_k(u_k,v_k)$ need to be obtained first. A typical scaling is inverse depth scaling with the following function [24]:

$$z_k(u_k,v_k) = \frac{1}{d_k(u_k,v_k)\cdot\left(\left(1/z_{k,\text{near}}\right)-\left(1/z_{k,\text{far}}\right)\right)+\left(1/z_{k,\text{far}}\right)}, \tag{4}$$

where the depth data $d_k(u_k,v_k)$ was originally normalized to the range $[0\cdots1]$ and $z_{k,\text{near}}$ and $z_{k,\text{far}}$ are the minimum and maximum depth values of the 3D scene, respectively.

In the next step, the 3D point is forward projected into the intermediate view. Combining both projections, the point-to-point homography can be written as follows:

$$\begin{pmatrix}u_\lambda\\ v_\lambda\\ z_\lambda(u_\lambda,v_\lambda)\end{pmatrix} = P_\lambda P_k^{-1}\begin{pmatrix}u_k\\ v_k\\ z_k(u_k,v_k)\end{pmatrix}. \tag{5}$$

Note that this notation differs from the general plane-to-plane homography formulation, since the depth values z_k and z_λ are maintained in (5) for one-to-one mapping between 2D image plane and 3D world coordinates. This mapping is carried out for all pixel positions (u_k,v_k) from view k. For obtaining the color value at a certain position (u_λ,v_λ) in the intermediate view, all color values $l_k(u_k,v_k)$ from view k that map onto position (u_λ,v_λ) are collected. Next, the front-most pixel with minimum projected depth $z_{\min,\lambda,k}$ is selected as follows:

$$\begin{aligned}&z_{\min,\lambda,k}(u_\lambda,v_\lambda)\\ &\quad= \min_{\forall u_k,v_k}\left\{ z_{\lambda,k,u_k,v_k}(u_\lambda,v_\lambda)\;\middle|\;\begin{pmatrix}u_\lambda\\ v_\lambda\\ z_\lambda(u_\lambda,v_\lambda)\end{pmatrix} = P_\lambda P_k^{-1}\begin{pmatrix}u_k\\ v_k\\ z_k(u_k,v_k)\end{pmatrix}\right\}.\end{aligned} \tag{6}$$

Depending on the 3D scene structure, the number of pixels from view k that map onto position (u_λ,v_λ) can vary and refer to the following cases:

(i) 0 pixel: disocclusion in intermediate view;

(ii) 1 pixel: regular projected content;

(iii) $2\cdots N$ pixel: occlusion.

For the color projection, the associated position $(u_{k,\min}, v_{k,\min})$ in the original view is required as follows:

$$\begin{aligned}&(u_{k,\min},v_{k,\min})\\ &\quad= \arg\min_{\forall u_k,v_k}\left\{ z_{\lambda,k,u_k,v_k}(u_\lambda,v_\lambda)\;\middle|\;\begin{pmatrix}u_\lambda\\ v_\lambda\\ z_\lambda(u_\lambda,v_\lambda)\end{pmatrix} = P_\lambda P_k^{-1}\begin{pmatrix}u_k\\ v_k\\ z_k(u_k,v_k)\end{pmatrix}\right\}.\end{aligned} \tag{7}$$

This position finally determines the color contribution $l_{\lambda,k}(u_\lambda,v_\lambda)$ from view k in the intermediate view:

$$l_{\lambda,k}(u_\lambda,v_\lambda) = l_k(u_{k,\min},v_{k,\min}). \tag{8}$$

The above process from (5) to (8) is repeated for view n to obtain the color contribution $l_{\lambda,n}(u_\lambda,v_\lambda)$:

$$l_{\lambda,n}(u_\lambda,v_\lambda) = l_n(u_{n,\min},v_{n,\min}). \tag{9}$$

Combining the contributions in both views, the general intermediate view interpolation between original views k and n can be formulated as follows:

$$\begin{aligned}l_\lambda(u_\lambda,v_\lambda) = (1-\lambda)\cdot l_k(u_{k,\min},v_{k,\min})\\ +\lambda\cdot l_n(u_{n,\min},v_{n,\min}),\end{aligned} \tag{10}$$

where the final color value $l_\lambda(u_\lambda,v_\lambda)$ is interpolated from the two projected color values $l_k(u_{k,\min},v_{k,\min})$ and $l_n(u_{n,\min},v_{n,\min})$ with minimum projected depth values from both views. For real data this general mathematical description needs to be refined to account for incorrect input data, for example, erroneous depth values at object boundary pixels, as shown in Section 4.2. In the following implementation of layered intermediate view synthesis, we omit all pixel position indices (u,v) for color and depth data for simplification, if they do not differ from the general case, shown in Section 3.

4. IMPLEMENTATION OF LAYERED INTERMEDIATE VIEW SYNTHESIS

After specifying the general projection process in Section 3, the adaptation toward real data is described here. Please note that in classical 2D video applications backward projection is used, where for each target pixel in the intermediate image the corresponding source pixels in the original images are sought. In 3D, however, this process becomes very complex, since many pixels from very different regions of the original images may map onto the target pixel such that original images have to be searched entirely to identify all possible source pixels. Therefore, a forward projection is applied here and the introduced holes are filled appropriately. The 3DV concept presented in Section 2 relies on the availability of high-quality intermediate view synthesis algorithms at the receiver. Previous approaches on view synthesis have concentrated on simple concepts without adequate occlusion handling [20, 25–27] or generate a point-based representation [28]. However, interpolation artifacts may result in unacceptable quality. In the example in Figure 1, for instance, from position 2 only virtual views are visible. A typical camera distance in a stereo setup is 5 cm. This means that original views V1 and V5 span 20 cm, a distance that is difficult to handle with DIBR. Severe artifacts are known to occur especially along object borders with large depth discontinuities. On the other hand, areas with smooth depth variations can be projected very reliably to virtual intermediate views. This implies separate processing of depth discontinuities and smooth depth regions. Depth discontinuities can be found easily within the depth images using edge detection algorithms.

Hence, our view synthesis process consists of three parts: layer extraction (edge detection and separation into reliable and boundary regions), layer projection (separate DIBR of regions and fusion), and intermediate view enhancement (correction, cleanup, and filtering). An overview of the process is shown in Figure 2. Input data for our method are original color and per-pixel depth data. The solid arrows represent color processing, while dashed arrows show depth processing or depth data usage for projection or edge detection purposes. From the depth information, the layers are extracted along the significant depth discontinuities, as described in Section 4.1. In the next stage in Figure 2, all layers from the marked color buffers are projected into separate layer buffers for the intermediate view. The intermediate view is created by merging the two projected main layers first. Afterwards, foreground and background boundary layers are added, as described in Section 4.2. Finally, image enhancement, such as hole filling and edge smoothing, are applied to create the final intermediate view, as shown in Section 4.3. The processing time of the algorithms depends linearly on the number of pixels in an image. That is, if image resolution is doubled, four times the processing time is required.

The idea to work with a layered approached was already investigated in [29] for the application of free viewpoint navigation, where a boundary layer of a certain width along significant depth discontinuities was extracted. In our approach, we further improved this idea. Moreover, while the approach in [29] operates with geometric primitives (triangles) for rendering, supported by 3D graphics functions, our approach was generically implemented as an image-based 3D warping process. Thus, we can actively control the different interpolation functions that occur in the view synthesis.

In computer graphics, such projection methods are sometimes implemented as point splat algorithms, where each pixel is defined as a 3D sphere with a certain radius, controlled by the point splat function. In such applications, interpolation and small hole filling are carried out automatically around each pixel, depending on the applied point splat function. Usually, this function is defined globally for an image, such that different requirements on hole filling and interpolation cannot be addressed. Therefore, we decided to use classical image-based interpolation algorithms to solve these problems and to improve the visual quality of synthesized views as described in Section 3.

4.1. Layer extraction

In the first part of the rendering approach, we distinguish between reliable and unreliable depth regions in the original views. The areas along object boundaries are considered unreliable, since boundary samples usually have mixed foreground/background colors and can create artifacts after projection into novel views. Further, errors from depth estimation mainly distort object boundaries. Therefore, similar to [29], significant depth discontinuities are detected to create main and boundary layers. For this, we use a Canny edge detector [30] with a content-adaptive significance threshold (110 in our experiments) operating on the depth images and mark a 7-sample-wide area as unreliable along the detected edges. The significance threshold value was found experimentally for the used test sets to give the best results in finding true depth edges. Since test data with appropriate depth maps is still very limited, further investigations on automatic threshold selection can only be carried out in the future, if more test data becomes available.

In contrast to [29], the unreliable area is split into a foreground and background boundary layers, as shown in Figure 3 as black and white areas, respectively, to allow different processing.

4.2. Layer projection

The layer projection extends the general formulation of depth-based intermediate view synthesis, presented in Section 3. This second part of the processing chain is the main block of the view synthesis algorithm. Inputs are a left and a right original images, associated depth maps, associated camera calibration information, the interpolation parameter $\lambda \in [0 \cdots 1]$, all presented in Section 3, and associated label information as shown in Figure 3. Differently labeled regions from both input images are projected to the virtual view position separately and the results are fused following depth ordering and reliability criteria.

Following the general approach, presented in Section 3, both main layers are projected into separate color or color

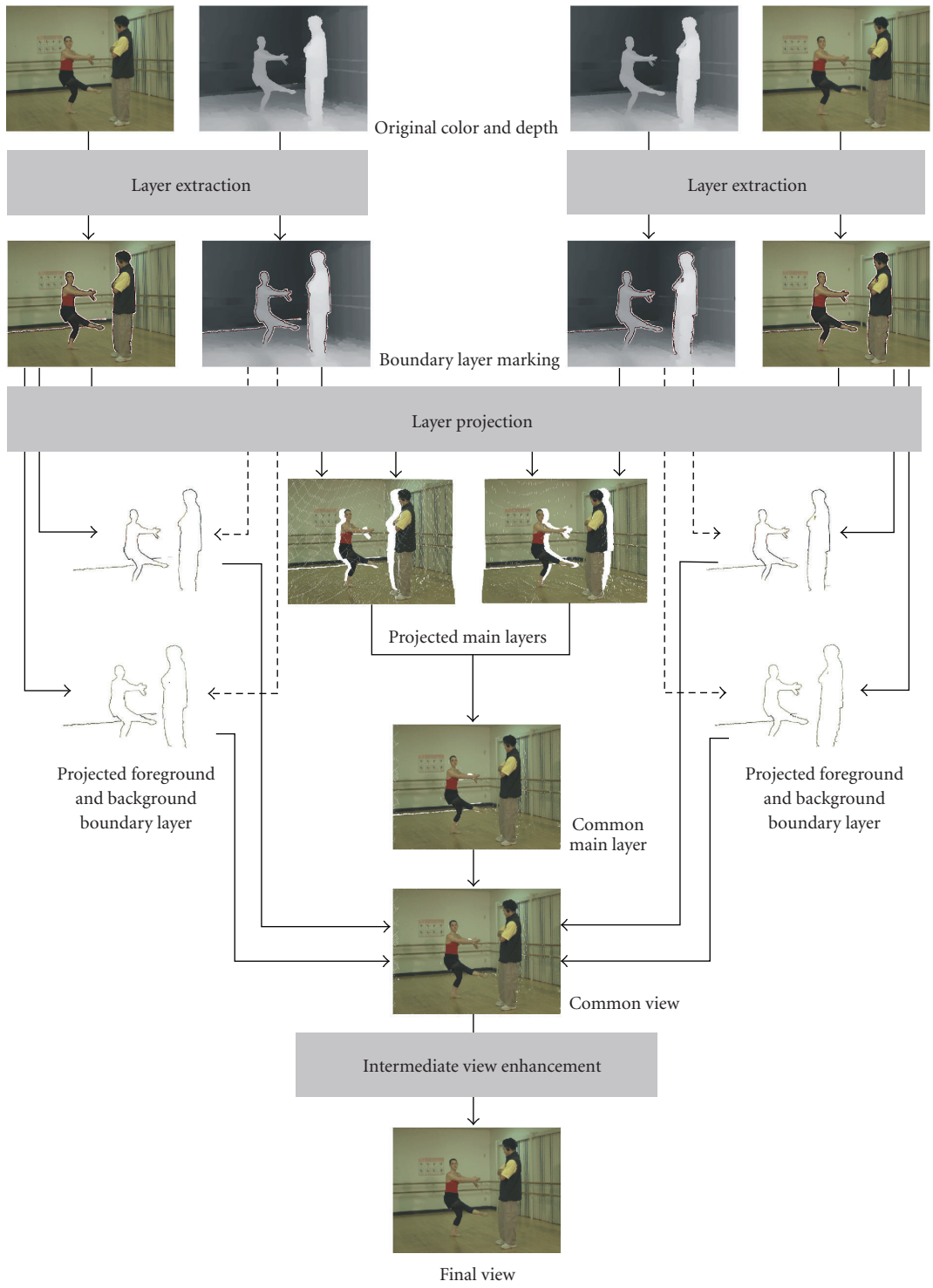

Figure 2: Structural overview of the proposed synthesis method.

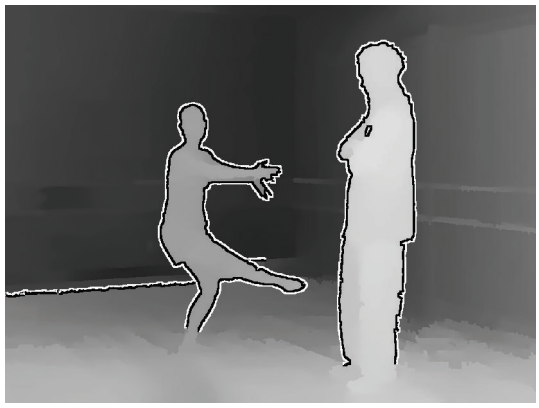

Figure 3: Layer Assignment along significant depth discontinuities: foreground boundary layer (black), background boundary layer (white), and main layer (grey values).

Figure 4: Common main layer after projection.

buffers l_1 and l_2, using the corresponding floating-point real depth data z_1 and z_2. From this, a common main layer $l_{M,\lambda}$ is created by varying the general interpolation formula (10) as follows:

$$l_{M,\lambda} = \begin{cases} (1-\lambda)l_1 + \lambda l_2, & \text{if } z_{\lambda,1}, z_{\lambda,2} \text{ exist, } |z_{\lambda,1} - z_{\lambda,2}| < \varepsilon, \\ l_2, & \text{if } z_{\lambda,1} \text{ does not exist or } z_{\lambda,2} > z_{\lambda,1} + \varepsilon, \\ l_1, & \text{if } z_{\lambda,2} \text{ does not exist or } z_{\lambda,1} > z_{\lambda,2} + \varepsilon, \end{cases} \tag{11}$$

where ε represents a significance value, which was set to 1.0 for the experiments and $z_{\lambda,1}$ and $z_{\lambda,2}$ represent the projected depth values with respect to the intermediate view. These projected depth values are used to decide on the depth ordering of both color values. The method in (11) guarantees that either the front-most sample from each view is used, or both samples are λ-interpolated, if they have similar projected depth values. The interpolation further reduces possible illumination differences between the original views and provides smooth transition when navigating from one original camera view to the other. A resulting common main layer is shown in Figure 4. The interpolation process (11) also creates a common floating-point depth buffer $z_{M,\lambda}$:

$$z_{M,\lambda} = \min(z_{\lambda,1}, z_{\lambda,2}). \tag{12}$$

In the next step, the foreground boundary layers $l_{F,1}$ and $l_{F,2}$ are projected and a common layer for color $l_{F,\lambda}$ and floating-point depth $z_{F,\lambda}$ is created similar to the main-layer method, described in (12). Then, the common main and foreground boundary layers are merged as follows:

$$l_{FM,\lambda} = \begin{cases} l_{F,\lambda}, & \text{if } z_{F,\lambda} \le z_{M,\lambda}, \\ l_{M,\lambda}, & \text{if } z_{F,\lambda} > z_{M,\lambda}. \end{cases} \tag{13}$$

Here, only a simple depth test is used. The front-most sample from either layer is taken, which is mostly the foreground boundary sample. Besides the new common color layer $l_{FM,\lambda}$, the associated depth layer $z_{FM,\lambda}$ is created similarly to (12).

In the last step of the projection process, the background boundary layers $l_{B,1}$ and $l_{B,2}$ are projected to $l_{B,\lambda}$ and merged with $l_{FM,\lambda}$:

$$l_{\lambda} = \begin{cases} l_{FM,\lambda}, & \text{if } z_{FM,\lambda} \text{ exists,} \\ l_{B,\lambda}, & \text{if } z_{FM,\lambda} \text{ does not exist} \end{cases} \tag{14}$$

to create the final color or color l_λ and depth z_λ similar to (12). The background layer information $l_{B,\lambda}$ is only used to fill empty regions in the intermediate view. Since the common main layer $l_{FM,\lambda}$ already covers most of the samples around foreground objects, as can be seen in Figure 4, only few background boundary samples are used and thus the color-distorted samples at object boundaries from the original views are omitted. Those are known to create corona-like artifacts within background regions using simple 3D warping algorithms, which is avoided by our layered approach with 2 different kinds of boundary layers. The result after layer projection is shown in Figure 5.

Figure 5: Intermediate view after layer projection.

4.3. Intermediate view enhancement

The last part of our algorithm provides postprocessing after layer projection and includes correction, cleanup, and

filtering processes. Two types of holes may still occur in the rendered images at this stage: small cracks and larger missing areas. The first type of holes is small cracks which can occur in the entire image area and are introduced by the forward mapping nature of image-based 3D warping. Each point from an original image is projected separately into the intermediate view, and falls in general onto a floating point coordinate. This position is quantized to the nearest neighbor position of the integer sample raster. Unfortunately, quantization may leave some samples unfilled being visible as thin black lines in Figures 4 and 5. In some cases, such cracks in foreground regions are filled by background information from the other original image. This results in artifacts as shown in Figure 6, *left*, where background samples shine through the foreground object.

Such artifacts are detected by finding depth values that are significantly larger than both neighboring values in horizontal, vertical, or diagonal directions:

$$\begin{aligned} g_{\text{hor}} &= 2\cdot z_\lambda(u_\lambda, v_\lambda) - z_\lambda(u_\lambda - 1, v_\lambda) - z_\lambda(u_\lambda + 1, v_\lambda), \\ g_{\text{ver}} &= 2\cdot z_\lambda(u_\lambda, v_\lambda) - z_\lambda(u_\lambda, v_\lambda - 1) - z_\lambda(u_\lambda, v_\lambda + 1), \\ g_{\text{diag},1} &= 2\cdot z_\lambda(u_\lambda, v_\lambda) - z_\lambda(u_\lambda - 1, v_\lambda - 1) - z_\lambda(u_\lambda + 1, v_\lambda + 1), \\ g_{\text{diag},2} &= 2\cdot z_\lambda(u_\lambda, v_\lambda) - z_\lambda(u_\lambda + 1, v_\lambda - 1) - z_\lambda(u_\lambda - 1, v_\lambda + 1). \end{aligned} \tag{15}$$

This refers to background pixels within a foreground region. From the directional significance values, the maximum value $g_{\max}$ is calculated as follows:

$$g_{\max} = \max\left(g_{\text{hor}}, g_{\text{ver}}, g_{\text{diag},1}, g_{\text{diag},2}\right). \tag{16}$$

If $g_{\max}$ exceeds a specific threshold (40 in our experiments), the color value $l_\lambda(u_\lambda, v_\lambda)$ is substituted by the median value of neighboring color values assuming that they have correct depth values assigned. The correction of such an artifact is also shown in Figure 6, *left*. Again, the specific threshold was determined experimentally and future investigations have to be carried out if new test data becomes available.

The second type of holes includes larger missing areas. They either occur due to erroneous depth values, or are areas that become visible in the intermediate view, while being occluded in both original views. Such larger holes are currently filled linewise with neighboring available background information, as shown in Figure 6, *middle*. Here, the two corresponding depth values at the two-hole boundary pixel are analyzed to find background color samples to extrapolate into the hole region. This simple constant-color extrapolation of the background pixel leads to better results, than an unconstrained linear interpolation between both values. Often, one of the hole boundary pixels belongs to the foreground object and its color value would lead to color bleeding into the hole. This approach leads to good filling results for missing areas due to depth errors. In cases of fillings for disocclusions, sometimes both hole boundary pixels are foreground pixels and the foreground color is incorrectly extrapolated into the background hole.

Here, one of the fundamental problems of view interpolation from sparse views occurs, which are disocclusions in intermediate views, where no original information is available in any view. For this, no general solution exists. In some cases, hole filling algorithms could be extended into the temporal dimension to hope for additional data in previous or future frames, if a foreground object has moved enough to reveal required background information. However, since the degree of motion cannot be predicted, this approach has limitations and was not considered for our implemented method.

Finally, foreground objects are low-pass filtered along the edges to provide a natural appearance, as shown in Figure 6, *right*. In the original views, object boundary samples are a color mixture of foreground-background due to initial sampling and filtering during image capturing. In the rendered intermediate views of our layered approach, these mixed color samples are often excluded in order to avoid corona artifacts in background areas. Consequently, some foreground-background boundaries look unnaturally sharp, as if foreground objects were artificially inserted into the scene. Therefore, the above-mentioned Canny edge detection filter [30] is applied to the final depth information z_λ of the intermediate view to detect edges with depth gradients $|\nabla z_\lambda|$ above the Canny significance threshold η ($\eta = 50$ in our experiments). Then, the color buffer is convolved with an averaging three-tap low-pass filter in both spatial directions at corresponding significant depth edges to provide a more natural appearance:

$$l_{\lambda,\text{Final}} = \begin{cases} l_\lambda * \dfrac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, & \text{if } |\nabla z_\lambda| \geq \eta, \\ l_\lambda, & \text{if } |\nabla z_\lambda| < \eta. \end{cases} \tag{17}$$

Additionally, the filtering helps to reduce remaining artifacts along depth discontinuities.

5. VIEW SYNTHESIS EXAMPLES

A resulting intermediate view after all processing steps is shown in Figure 7.

Here, the middle view between two original cameras was synthesized, that is, $\lambda = 0.5$, which corresponds to a physical distance of 1 cm to both original cameras in this case. The virtual view is of excellent quality without visible artifacts.

Details of rendered views are shown in Figure 8. The top row shows examples of standard 3D warping without the specific layer projection steps introduced in Section 4. Corona artifacts occur at foreground/background boundaries. Some dark foreground pixels are mistakenly added to lighter background areas, resulting in typical corona-type additional contours around objects. Such artifacts can change over time, resulting in very annoying effects within the rendered video. This can make the whole concept of 3DV unacceptable. The bottom row in Figure 8 shows the corresponding rendering details using our improvements to the 3D warping process as introduced in Section 4. Corona artifacts are widely removed. With minimum artifacts of individual images, also the video quality is significantly increased, thus our views synthesis algorithm is capable of

Figure 6: Artifacts (*top*) and artifact removal (*bottom*). Crack sample removal (*left*), area filling (*middle*), and edge smoothing (*right*).

Figure 7: Final intermediate view synthesis after filtering.

forming the basis for the advanced 3DV concept based on MVD.

Further comparisons, for example, with the method from [29] are only limited, since for this method, no results for the Ballet sequence are available. For the Breakdancers sequence, the interpolation quality seems to be equal, although different algorithms were applied. In our approach, we used a fixed boundary layer width, while in [29], complex alpha matting is used to deal with semitransparent areas. Also no hole filling was applied in [29], such that a comparison is difficult. Future test data with complex depth structure will show whether one method has advantages over the other. Currently, we achieve very good visual results with our synthesis method.

The purpose of the view interpolator is to create N input views for a 3DV system out of M views plus depth of an MVD representation. One example is the Philips auto-stereoscopic display, where 9 views with eye-distance (approx. 5 cm) are required as input. For such a setup, as illustrated in Figure 1, five of the resulting nine views are shown in Figure 9 for the Ballet and Breakdancers datasets. The camera spacing of these datasets is 20 cm. Three intermediate views with $\lambda = \{1/4, 1/2, 3/4\}$ have been created between two original cameras. The leftmost and rightmost images in Figure 9 are original views. The three images inbetween are virtual views not exhibiting any artifacts. Pairwise stereoscopic views are available to support motion parallax and 3D depth impression.

6. CONCLUSIONS

An advanced system for 3DV based on MVD is presented in this paper. It efficiently supports auto and multiview stereoscopic displays. This latter type of 3D displays enables multiuser 3DV sensation in a living room environment without the necessity to wear glasses, but with motion parallax impression and full social interaction. MVD can serve as a generic format for 3DV in this concept as it has clear advantages over alternative concepts based on MVC or MPEG-C Part 3 in terms of data rate, quality, and functionality. This concept, however, integrates a number of sophisticated processing steps that partially still require research. Among those, high-quality intermediate view synthesis is crucial to make this concept feasible. It is known that such algorithms may introduce annoying artifacts along depth discontinuities. Therefore, the approach presented here separates input images in reliable and unreliable areas based on edge detection in high-quality depth images, since these edges correspond to depth discontinuities. Reliable and unreliable image areas are treated separately and the results are merged depending on reliability criteria. Specific postprocessing algorithms are introduced to further enhance rendered view quality. This includes different hole-filling approaches as well as a final smoothing filter along depth discontinuities in the rendered views to reduce remaining

Figure 8: Details in the intermediate view for simple merging and our proposed method.

Figure 9: Five views in stereo pair distance for 9-view auto-stereoscopic display. Two views at original camera positions (*far left* and *far right*) and intermediate views for Ballet (*top*) and Breakdancers dataset(s) (*bottom*).

artifacts. A position-dependent blending factor is used to weight contributions from different input images. The presented results show that the processing in layers taking reliability information along depth discontinuities into account significantly reduces rendering artifacts. Corona artifacts that frequently occur with standard 3D warping are widely eliminated. High-quality intermediate views are generated with the presented algorithm. With this, an important building block within the advanced 3DV concept for MVD is shown to be available. Besides further optimization, our future work will include development of all other building blocks such as acquisition, depth estimation, coding, and transmission as well as the final system integration.

ACKNOWLEDGMENT

The authors would like to thank the Interactive Visual Media Group of Microsoft Research for providing the *Ballet* and *Breakdancers* datasets.

REFERENCES

[1] A. Smolic, K. Mueller, P. Merkle, et al., "3D video and free viewpoint video—technologies, applications and MPEG standards," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 2161–2164, Toronto, Canada, July 2006.

[2] J. Konrad and M. Halle, "3-D displays and signal processing: an answer to 3-D ills?" *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 97–111, 2007.

[3] P. Kauff, N. Atzpadin, C. Fehn, et al., "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication*, vol. 22, no. 2, pp. 217–234, 2007.

[4] October 2008, http://www.philips.com/3Dsolutions.

[5] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, 2007.

[6] C. Fehn, P. Kauff, M. Op de Beeck, et al., "An evolutionary and optimised approach on 3D-TV," in *Proceedings of the International Broadcast Convention (IBC '02)*, pp. 357–365, Amsterdam, The Netherlands, September 2002.

[7] ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC FDIS 23002-3 Representation of Auxiliary Video and Supplemental Information," Doc. N8768, Marrakech, Morocco, January 2007.

[8] ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC 13818-1:2003/FDAM2 Carriage of Auxiliary Data," Doc. N8799, Marrakech, Morocco, January 2007.

[9] F. Forster, M. Lang, and B. Radig, "Real-time range imaging for dynamic scenes using colour-edge based structured light," in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR '02)*, vol. 3, pp. 645–648, Quebec, Canada, August 2002.

[10] J. Salvi, J. Pages, and J. Battle, "Patter codification strategies in structured light systems," *Pattern Recognition*, vol. 37, no. 4, pp. 827–849, 2004.

[11] R. Koch, M. Pollefeys, and L. Van Gool, "Multi viewpoint stereo from uncalibrated video sequences," in *Proceedings of the 5th European Conference on Computer Vision (ECCV '98)*, vol. 1406 of *Lecture Notes in Computer Science*, p. 55, Springer, Freiburg, Germany, June 1998.

[12] Y. Li, C.-K. Tang, and H.-Y. Shum, "Efficient dense depth estimation from dense multiperspective panoramas," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 119–126, Vancouver, Canada, July 2001.

[13] C. Strecha, T. Tuytelaars, and L. Van Gool, "Dense matching of multiple wide-baseline views," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1194–1201, Nice, France, October 2003.

[14] J. Woetzel and R. Koch, "Real-time multi-stereo depth estimation on GPU with approximative discontinuity handling," in *Proceedings of the 1st European Conference on Visual Media Production (CVMP '04)*, pp. 245–254, London, UK, March 2004.

[15] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[16] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, Mass, USA, 1993.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2003.

[18] K. Shoemake, "Animating rotation with quaternion curves," *ACM SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 245–254, 1985.

[19] S. Avidan and A. Shashua, "Novel view synthesis in tensor space," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 1034–1040, San Juan, Puerto Rico, USA, June 1997.

[20] S. E. Chen and L. Williams, "View interpolation for image synthesis," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, pp. 279–288, Anaheim, Calif, USA, August 1993.

[21] J. S. McVeigh, M. W. Siegel, and A. G. Jordan, "Intermediate view synthesis considering occluded and ambiguously referenced image regions," *Signal Processing: Image Communication*, vol. 9, no. 1, pp. 21–28, 1996.

[22] D. Scharstein, "Stereo vision for view synthesis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 852–858, San Francisco, Calif, USA, June 1996.

[23] G.-C. Chang and W.-N. Lie, "Multi-view image compression and intermediate view synthesis for stereoscopic applications," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '00)*, vol. 2, pp. 277–280, Geneva, Switzerland, May 2000.

[24] ISO/IEC JTC1/SC29/WG11, "Description of Exploration Experiments in 3D Video," Doc. N9596, Antalya, Turkey, January 2008.

[25] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura, "Viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images," in *Stereoscopic Displays and Virtual Reality Systems II*, S. Fisher, J. Merritt, and B. Bolas, Eds., vol. 2409 of *Proceedings of SPIE*, pp. 11–20, San Jose, Calif, USA, February 1995.

[26] M. Morimoto and K. Fujii, "A view synthesis method based on object coding and simple disparity model," in *Proceedings of the IEEE International Symposium on Communications and Information Technologies (ISCIT '04)*, vol. 2, pp. 853–856, Sapporo, Japan, October 2004.

[27] J. D. Oh, S. Ma, and C.-C. J. Kuo, "Disparity estimation and virtual view synthesis from stereo video," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '07)*, pp. 993–996, New Orleans, La, USA, May 2007.

[28] N. L. Chang and A. Zakhor, "A multivalued representation for view synthesis," in *Proceedings of the International Conference on Image Processing (ICIP '99)*, vol. 2, pp. 505–509, Kobe, Japan, October 1999.

[29] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.

[30] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.