Correspondence and Depth-Image Based Rendering
a Hybrid Approach for Free-Viewpoint Video

Christian Lipski, Felix Klose and Marcus Magnor

***Abstract*—We present a novel approach to free-viewpoint video. Our main contribution is the formulation of a hybrid approach between image morphing and depth-image based rendering (DIBR). When rendering the scene from novel viewpoints, we use both dense pixel correspondences between image pairs as well as an underlying, view-dependent geometrical model. Our novel reconstruction scheme iteratively refines geometric and correspondence information. By combining the strengths of both depth and correspondence estimation, our approach enables free-viewpoint video also for challenging scenes as well as for recordings that may violate typical constraints in multi-view reconstruction. For example, our method is robust against inaccurate camera calibration, asynchronous capture, and imprecise depth reconstruction. Rendering results for different scenes and applications demonstrate the versatility and robustness of our approach.**

***Index Terms*—free-viewpoint video, view interpolation, optical flow, depth reconstruction, multi-view data**

I. Introduction

Free-viewpoint video, i.e. the ability to re-render a recorded scene from novel viewpoints, has been a long-standing research topic [9]. Often, synchronized cameras and elaborate recording setups are used to produce high-quality free-viewpoint video [10], [18], [38]. Using state-of-the art geometry reconstruction algorithms, the input material can be re-rendered from novel viewpoints. In order to do so, one must make strong assumptions about the captured data. Typically, input images have to be captured synchronously or scene content must remain static.

For input material that violates these assumptions, other solutions have to be found. Recently, free-viewpoint systems have been introduced that use optical flow or dense correspondence information to improve the rendered result [11], [12], [23]. Although these approaches can cope with a wider range of input data, other problems arise: Finding general 2D image correspondences is a much harder problem than finding matches that obey the epipolar constraint. If the depth is unknown, occlusion reasoning becomes an additional problem. Depending on the rendering algorithm, view synthesis might be constrained to the visual hull of the original cameras.

We present a rendering scheme that aims to combine the advantages of both conventional depth-based as well as image warping-based approaches. The key idea is to apply image morphing in 3D space: We expect geometric scene information to be approximate, faulty or even missing. Therefore, we compensate for inaccuracies and invalid assumptions made in reconstruction by aligning image regions according to bidirectional correspondence maps.

Institut für Computergraphik, TU Braunschweig, GERMANY.
Manuscript received 07/09/2013

After we give a brief overview of the state of the art in Sect. II, we discuss the basic properties of warping based rendering and DIBR in Sect. III. In Sect. IV we introduce our first main contributions: We present a hybrid rendering equation that incorporates both image warping and DIBR, cf. Fig. 1 (right). This enables us to significantly reduce rendering artifacts in scenes that violate common assumptions made in DIBR (e.g., static scene content, epipolar constraint). Our second main contribution is the iterative scheme for joint reconstruction of image correspondences and depth, cf. Sect. V. The key element in our reconstruction is a soft geometric constraint that enforces correspondences to be consistent with a geometric proxy. From these correspondence maps, robust, dense depth maps are derived. The geometric constraint is updated accordingly and correspondence maps are re-estimated, cf. Fig. 2.

In Sect. VI, the technical details of the rendering pipeline are presented. After showcasing different data sets and applications in Sect. VII, we conclude in Sect. IX.

II. Related Work

Novel capturing techniques along with progress in reconstruction and rendering algorithms led to remarkable progress in free-viewpoint video: Dense arrays of synchronized cameras are an established way to automatically create depth-based free-viewpoint video [38]. Joint segmentation and reconstruction enables wide-baseline captures while maintaining a high visual quality [18]. High-resolution laser scans of performing actors also considerably improve reconstruction and rendering [10]. Similarily, range sensors or user-created geometry may improve optical flow computation [29]. By using precisely triggered lighting and capturing devices, relighting can be incorporated into light-field based free-viewpoint video [11]. For small viewpoint alterations, (dis-)occlusion handling can be elegantly circumvented by nonlinear disparity mapping [20]. Unfortunately, this technique is not suitable for rendering actual transitions between images.

An extensive survey on free-viewpoint video has been conducted by Smolic [30], a juxtaposition of capture stages and algorithms was presented by Starck et al. [32]. We would like to focus on those publications that successfully cope with so-called “casually” captured material, i.e., unsynchronized digital single-lens reflex (DSLR) and camcorder recordings. We identified two recent trends in dealing with this kind of input material. One approach is to incorporate automatic corrections or user-specified cues into reconstruction and rendering. Another way to tackle difficult scenes is to use an approximate geometric representation of the scene instead of trying to reconstruct a detailed but faulty geometry.

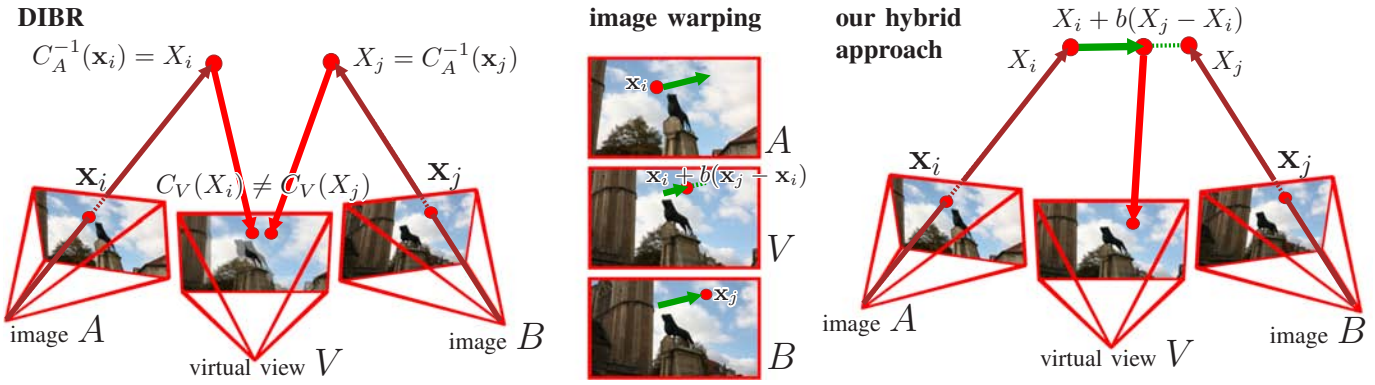

Fig. 1: Correspondence and depth-image based rendering. In conventional depth-image based rendering (DIBR, left), corresponding pixel positions $\mathbf{x}_i$, $\mathbf{x}_j$ in images A, B are projected to their estimated world space positions X_i, X_j. Due to reconstruction inaccuracies and invalid assumptions, reprojections of X_i, X_j might not align in a virtual view V. In image morphing (middle), pixel correspondences are directly used to create an in-between view V. In our proposed approach (right), we apply DIBR to reconstruct world space positions and use estimated correspondences to align corresponding points X_i, X_j in world space.

Correction-based approaches: On challenging urban scenes, cues for object boundaries and scene depth can help significantly to improve the rendering quality of depth-image based rendering [7], [8]. Alternatively, image correspondences can be used as a basis for warping-based rendering, as Chen and Williams presented in their seminal paper [9]. Their rendering formulation has recently been used for dynamic, real-world recordings [22], [23]. To achieve visually convincing renderings, faulty image correspondence maps have to be manually corrected [19]. For small local reprojection errors in multi-view rendering of human actors, real-time optical flow can help to automatically realign the different projections [12]. Incorporating the correction step directly into the reconstruction stage, we handle larger scenes with more complex occlusions and disocclusions. With our hybrid reconstruction and rendering scheme we can often avoid artifacts with no or significantly less user interaction than previous approaches.

Approximative representations: To browse through huge photo collections, simple billboards proved to be a good approximation [31]. Alternatively, non-reconstructable parts of the images can be represented with a randomized point cloud [17]. For many dynamic scenes with known or reconstructable background geometry it can be beneficial to handle background and foreground differently. Billboard approximations of moving actors have been used successfully for navigating through multi-view data of moving actors [2]. For free-viewpoint video football replays, a billboard cloud has been used to approximate the individual players [15], [16]. For such data sets, it is often infeasible to reconstruct an accurate per-pixel depth. We pick up on the idea of approximating hard-to-reconstruct regions of images by a set of planar surfaces. Our hybrid approach makes it possible to replace commonly used cross-blending of approximative geometry by pixel-exact warping of visually similar regions.

III. Background

Let us briefly revise the unique strengths and features of both depth-image as well as warping based rendering. In depth-image based rendering (DIBR) [13], for any given pixel location (u, v) in an image A, not only the color information $I_A(u, v) = (r, g, b)$, but also depth information $D_A(u, v) = d$ is available. The image space location $\mathbf{x}$ in an image A is defined as $\mathbf{x} = (u, v, D_A(u, v), 1)$. Along with intrinsic and extrinsic camera parameters, the original 3D location of the pixel can be reconstructed, as the depth $D_A(u, v)$ is known.In order to synthesize the view of a virtual image V, each pixel of an image A at a given location $\mathbf{x}$ has to be reprojected to the image plane of V.

$$\tilde{I}_A(C_V({C_A}^{-1}(\mathbf{x}))) = I_A(\mathbf{x}) \tag{1}$$

C_A denotes the projection from 3D world space to the image space of image A and ${C_A}^{-1}$ is the inverse projection, cf. Fig. 1 (left). C_A is defined by

$$C_A(X) = \mathbf{A}_A \mathbf{P}_n \mathbf{D}_A(X) \tag{2}$$

where X is a world space point in homogeneous coordinates, $\mathbf{D}_A$ and $\mathbf{A}_A$ are the extrinsic and intrinsic matrices associated to image A and $\mathbf{P}_n$ is the normalized perspective projection matrix [35].

Typically, more than one image is used for image synthesis, e.g., the rendered image can be a combination of two reprojected images A, B. In order to render V, its extrinsic and intrinsic camera parameters $\mathbf{D}_V$ and $\mathbf{A}_V$ as well as weighting coefficients a, b with $a + b = 1, \quad 0 \leq a, b \leq 1$ have to be provided. Both input images are reprojected according to (1) and blended according to weights a, b.

$$I_V(\mathbf{x}) = a\tilde{I}_A(\mathbf{x}) + b\tilde{I}_B(\mathbf{x}) \tag{3}$$

Another possibility to render in-between views is image morphing (also known as image warping). In contrast to DIBR, image synthesis is done in image space, cf. Fig. 1 (middle). When rendering a virtual view V between images A and B, we

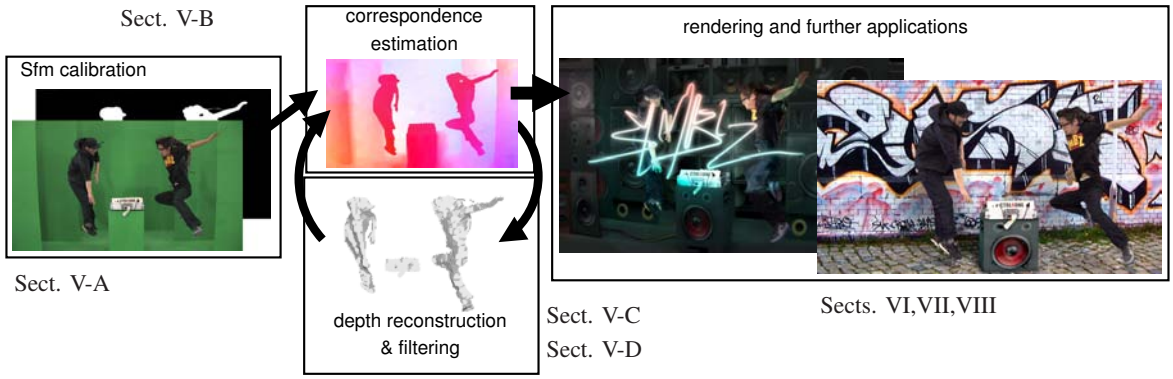

Fig. 2: Overview of reconstruction and rendering pipeline. After camera calibration and initial (sparse) depth reconstruction (Sect. V-A), correspondences are estimated for neighboring images (Sect. V-B). We reconstruct depth using triangulation (Sect. V-C). Since we expect depth inaccuracies and holes, we apply a filtering to the depth maps to obtain robust results(Sect. V-D). Correspondences are refined using reconstructed depth as a constraint. Both depth maps and correspondences are used for rendering (Sect. VI). We showcase our proposed approach on different data sets (Sect. VII) and applications (Sect. VIII), allowing composites of our free-viewpoint renderings with 3D objects and other camcorder captures.

assume that for every pixel location $\mathbf{x}$ in A, the corresponding location $\mathbf{x}+\mathbf{w}_{AB}(\mathbf{x})$ in B is known (and vice versa). $\mathbf{w}_{AB}$ is a so-called correspondence map which stores for each given pixel position $\mathbf{x}$ in A the vector pointing to the corresponding location in B. Since we represent V as a weighted combination of A and B, this correspondence information can be used to warp each pixel towards its corresponding counterpart:

$$\tilde{I}_A(\mathbf{x}+b\mathbf{w}_{AB}(\mathbf{x}))=I_A(\mathbf{x}) \tag{4}$$

It is important to understand the differences between both approaches. DIBR provides the ability to arbitrarily extrapolate viewpoints. Any virtual image V, with a camera orientation reasonably close to one of the original images, can be rendered. The availability of depth information can also be exploited to resolve occlusion ambiguities during rendering. DIBR makes strong assumptions regarding camera calibration, capturing modalities and scene appearance. Often, a precise calibration of cameras is needed. In addition, the scene content has either to remain static during acquisition, or multiple cameras have to be triggered synchronously. Another assumption is that visually identical image regions depict the same scene object in world space. This assumption can be violated, a prominent example are the silhouettes of a dolphin [6]. Although they can be matched between images taken from various angles, corresponding pixels do not depict the same location on a dolphin's surface. Specular materials lead to similar problems, since bright highlights move on object surfaces when observed from different perspectives. Classical 3D reconstruction, i.e. triangulation of viewing rays, will fail to reconstruct consistent depth in these cases. If search for corresponding pixels is confined to the epipolar line, it is also possible that the correct matches will not be obtained at all.

Image Morphing does not suffer from any of the above mentioned constraints. As long as a perceptually valid bipartite matching of image locations is found, a convincing view interpolation can be synthesized [33]. The reconstruction is also not influenced by inaccurate camera calibration or dynamic scene elements. The downside is that virtual views are usually confined by the convex hull of the recorded images, i.e., no arbitrary extrapolation of camera positions is possible. Although warping weights < 0 and > 1 can be chosen, view extrapolation is still limited to camera positions that are a linear combination of the original input images. E.g., when the cameras are placed right next to each other (as in a stereoscopic setup), the virtual view can only move sideways, but not up, down, forward or backward. In cases where view locations locations are embedded into a lower-dimensional subspace, view interpolation is only possible in one or two dimensions [23]. Another drawback for morphing-based algorithms is occlusion handling which cannot be solved by the usual simple depth comparison. Furthermore, correspondence estimation is a 2D search problem and poses a less constrained and therefore more difficult problem then a 1D search along an epipolar line. Since the corresponding pixel location can be situated anywhere in the 2D domain of the captured image, the correspondence estimate is more ambiguous than the constraint search along the epipolar line. Furthermore, 2D correspondence estimation algorithms cannot exploit additional assumption made in (multi-view) stereo, e.g., many stereo algorithm explicitly reconstruct surface normals which allow for a more accurate appearance prediction in other views.

IV. A Hybrid Scene Model

In this section, we describe our hybrid approach of depth- and warping-based rendering that can cope with the specific limitations of the two separate techniques. In order to robustly handle real world scenes we specify our iterative scheme for depth and correspondence estimation in Sect. V. Technical details regarding the rendering will be discussed afterwards in Sect. VI.

Let us assume we captured two real world images A, B. We know camera parameters of A and B as well as the depth of every pixel. For a given pixel location $\mathbf{x}_i$, we can transform this location in image A to its world space position X_i using the DIBR equation (1), camera calibration and the

reconstructed depth. As discussed above (cf. Sect. III), the backprojected location in B might not depict the same object due to several reasons:

- Images/pixels not captured at the same time instant
- Non-Lambertian objects visible in scene
- Calibration errors
- Inaccurate Depth estimates
- Other model violations

Let us try to accurately project the pixel at position $\mathbf{x}_i$ in image A to its corresponding position $\mathbf{x}_j$ in image B, where $\mathbf{x}_j = \mathbf{x}_i + \mathbf{w}_{AB}(\mathbf{x}_i)$, cf. Eq. 4. The color values of the corresponding points should be visually similar:

$$I_A(\mathbf{x}_i) \sim I_B(\mathbf{x}_j) \tag{5}$$

Assuming knowledge of camera calibration, we can deduct $\mathbf{x}_i$ from its world space position X_i, as well as $\mathbf{x}_j$ from X_j.

$$I_A(C_A(X_i)) \sim I_B(C_B(X_j)) \tag{6}$$

Further substitution enables us to represent $\mathbf{x}_j$ as a transformation of

$$\begin{aligned} I_B(C_B(X_j)) &= I_B(C_B({C_B}^{-1}(\mathbf{x}_j))) \\ &= I_B(C_B({C_B}^{-1}(\mathbf{x}_i + \mathbf{w}_{AB}(\mathbf{x}_i)))) \end{aligned} \tag{7}$$

This means that knowing depth, camera calibration and image correspondences, we can obtain a valid transformation of every location in A to the image plane of B. In contrast to DIBR, we can eliminate the errors mentioned above as long as we know visually plausible image correspondences $\mathbf{w}_{AB}$. Similar to warping-based rendering, intermediate views I_V can be rendered by interpolating pixel positions. Instead of warping pixel location $\mathbf{x}_i$ towards $\mathbf{x}_j$ in image space, we transform pixel locations to world space positions X_i, X_j and determine the final pixel location by reprojecting the interpolated position to the image plane of V. While warping-based rendering only allows linear transitions between A and B, our scheme allows to reproject the images to an arbitrary virtual camera V:

$$\begin{aligned} &\tilde{I}_A(C_V(X_i + b\mathbf{W}_{AB}(\mathbf{x}_i))) = I_A(\mathbf{x}_i) \sim \\ &\tilde{I}_B(C_V(X_j + a\mathbf{W}_{BA}(\mathbf{x}_j))) = I_B(\mathbf{x}_j),\ a+b=1 \end{aligned} \tag{8}$$

Notice that $\mathbf{W}_{AB}(\mathbf{x}_i) = X_j - X_i$ and $\mathbf{W}_{BA}(\mathbf{x}_j) = X_i - X_j$. Practically, this means that we project every pixel $\mathbf{x}$ in A to its world space position X. According to the weight b, we move it towards the corresponding world space point of image B. Weight b is derived by a linear weighting function that takes the camera positions of A, B, V into account. We project every pixel into the image plane of V and repeat this process for all pixels of image B, cf. Fig. 1 (right). The result are two forward-warped images $\tilde{I}_A, \tilde{I}_B$. The final synthetic image is a blend of these two: $I_V(\mathbf{x}) = a\tilde{I}_A(\mathbf{x}) + b\tilde{I}_B(\mathbf{x})$. Our approach combines strengths of both approaches: Like DIBR we use geometric information (i.e. depth) to project image texture into world space and to resolve occlusion. Similar to image morphing, we also allow transitions between different source images by forward warping of pixels according to image correspondences. To make this hybrid approach possible, we apply these transitions in world space instead of image space. In contrast to DIBR, a plausible reprojection is possible despite imperfect depth reconstruction. This extends to the case where no depth can be reconstructed in parts of the image, in Sect. V we will explain how we can make it work with a rough depth estimate. In contrast to warping-based rendering, we can extrapolate the camera view arbitrarily and use the depth information to better handle occlusion. Also, a view-dependent surface model is created as a by-product that is valuable for further applications such as re-lighting, shadow casting and deep compositing.

V. Hybrid Reconstruction

In this section we will describe how we obtain an initial set of image correspondences, derive dense depth maps from these correspondences and use this depth information to compute a second generation of correspondences, cf. Fig. 2.

A. Structure from Motion (SfM) calibration

Using established SfM techniques, we estimate camera calibration [31] and (optionally) a sparse 3D point cloud representation of the scene [14], cf. Fig. 3 (top). For every pixel $\mathbf{x}_i$ in a reference image A, we query the 3D scene structure if a point X_i exists that projects to it: $|\mathbf{x}_i - C_A(X_i)|_\infty < 0.5\text{pixels}(px)$. If such a point exists, we can derive a correspondence vector estimate $\tilde{\mathbf{w}}_{AB}(\mathbf{x}_i)$ for any neighboring view B:

$$\tilde{\mathbf{w}}_{AB}(\mathbf{x}_i) = C_B({C_A}^{-1}(\mathbf{x}_i)) - \mathbf{x}_i \tag{9}$$

B. Correspondence Estimation

The initial geometric reconstruction can be exploited to constrain the correspondence search. For the reasons mentioned in Sect. IV, $\tilde{\mathbf{w}}_{AB}(\mathbf{x}_i)$ might differ from $\mathbf{w}_{AB}(\mathbf{x}_i)$, but for the set of points reconstructed in Sect. V-A, we assume that the difference is small: $\tilde{\mathbf{w}}_{AB}(\mathbf{x}_i) \sim \mathbf{w}_{AB}(\mathbf{x}_i)$, cf. Fig. 3. Our correspondence search is based upon a optimization scheme [21], [24] that minimizes an energy term $E = E_{data} + E_{smooth} + E_{symmetry}$. We add an additional geometric constraint E_{geom} that enforces consistency between the (sparse) geometric information and the estimated correspondences.

$$\begin{aligned} E(\mathbf{w}_{AB}) &= E_{data} + E_{smooth} + E_{symmetry} + E_{geom} \\ &= \sum_{\mathbf{x}} \|d_A(\mathbf{x}) - d_B(\mathbf{x} + \mathbf{w}_{AB}(\mathbf{x}))\|_1 && (10) \\ &+ \sum_{(\mathbf{x},\mathbf{y})\in\epsilon} \min(\alpha\|\mathbf{w}_{AB}(\mathbf{x}) - \mathbf{w}_{AB}(\mathbf{y})\|_1, d) && (11) \\ &+ \sum_{\mathbf{x}} \min(\alpha\|\mathbf{w}_{AB}(\mathbf{x}) + \mathbf{w}_{\mathbf{BA}}(\mathbf{x} + \mathbf{w}_{AB}(\mathbf{x}))\|_2, d) && (12) \\ &+ \sum_{\mathbf{x}} \min(\tilde{\alpha}\|\mathbf{w}_{AB}(\mathbf{x}) - \tilde{\mathbf{w}}_{AB}(\mathbf{x})\|_2, d) && (13) \end{aligned}$$

As proposed by Liu et al. [24], we want to ensure that matched pixels have a similar appearance. This is expressed by the data term E_{data}(10), where $d_A(\mathbf{x})$ is the (multi-dimensional) descriptor of a pixel location $\mathbf{x}$ in image A. We use the L^1-norm for descriptor matching as proposed by Lowe et al. [25]. We also enforce that all pairs of pixel pairs $(\mathbf{x}, \mathbf{y})$ contained in the set of pixel neighborhoods (ϵ)

have a similar correspondence vector in the smoothness term E_{smooth}(11). In order to reduce the computational complexity, we decouple the horizontal and vertical component of the smoothness term (L^1-norm). Parameters α, d are weighting and truncation values for the smoothness, symmetry and geometric term. The bidirectional symmetry term $E_{symmetry}$(12) for joint estimation of $\mathbf{w}_{AB}$ and $\mathbf{w}_{BA}$ is used as described by Lipski et al. [21]: Both correspondence fields $\mathbf{w}_{AB}$, $\mathbf{w}_{BA}$ are computed simultaneously. After each iteration of the global optimization, the intermediate result for $\mathbf{w}_{AB}$ is updated and used to evaluate the symmetric term of $\mathbf{w}_{BA}$ (and vice versa). Global convergence of symmetry is achieved after 20 iterative updates of the correspondence vectors. E_{geom}(13) is the newly created geometric term that enforces $\mathbf{w}_{AB}$ to coincide with the geometric reprojection $\tilde{\mathbf{w}}_{AB}$ according to the depth/scene flow model. We set $\tilde{\alpha} = \alpha$ for every pixel where depth information is available and $\tilde{\alpha} = 0$ elsewhere. We empirically determined that the L^2-norm produces pleasing results when used as a distance function for $E_{symmetry}$ and E_{geom}.

C. Depth Reconstruction

We triangulate an actual world space position $\tilde{X}_i$ if we know at least one valid corresponding pixel $\mathbf{x}_j$ in another image B by intersecting both viewing rays. We define correspondences as symmetric, if $\|\mathbf{w}_{AB}(\mathbf{x}_i)+\mathbf{w}_{BA}(\mathbf{x}_i+\mathbf{w}_{AB}(\mathbf{x}_i))\|_2 < e_{sym}$, where typically $e_{sym} = 3px$. Due to the many possible errors in depth estimation, we do not expect corresponding viewing rays to intersect in 3D space, cf. Fig. 3. Instead, we approximate $\tilde{X}_i$ as the point closest to all viewing rays in a least-squares sense. The depth associated with $\mathbf{x}_i$ is the z-component of $\tilde{X}_i$ projected into image space coordinates of A. The reprojection error $e_{repr}(\mathbf{x}_i)$, i.e. the image space distance between $\mathbf{x}_i$ and $C_A(\tilde{X}_i)$ indicates if the reconstructed depth is valid. In order to deal with imprecise calibration, inaccurate correspondences and asynchronous captures, we set the reprojection error threshold to a high value of $e_{max} = 10px$. We would like to point out that this error tolerant depth reconstruction leads to plausible depth values in image regions where traditional (multi-view) stereo reconstruction fails. The downside of this lean depth reconstruction are inaccurate or simply wrong depth estimates for spurious pixels. In order to cope with these inaccuracies, we perform an additional depth filtering step.

D. Depth Filtering

Since we expect the depth map to have holes (i.e. invalid pixels) and to include outliers (due to our high tolerance on reprojection error), we use a strong filtering to obtain a robust result: We assume piecewise-planar surfaces. Therefore, each image is segmented into super-pixels using the Simple Linear Iterative Clustering (SLIC) segmentation [1]. For the image sizes used in our experiments, we configure SLIC to obtain 1000 segments. For each superpixel, we use RANSAC to fit a plane to the reconstructed world space points, cf. Fig. 3 (bottom). Only these points contribute which are considered as valid. For super-pixels without a sufficient number of valid pixels, depth is simply assigned from averaging neighboring segments. We experimentally determined that at least 5% of a segment's pixels should be valid. Our filtering scheme effectively trades accuracy for robustness. All reconstructed depths are forced to be coplanar with all other pixels of the same SLIC segment. This approach is feasible due to our error-tolerant rendering scheme. Possible geometric inaccuracies introduced by the depth filtering are compensated by our hybrid scene model, cf. Sect. IV. Our depth filtering could possibly be used as a preprocessing step for other depth-based renderers, but would introduce new blurring or ghosting artifacts.

E. Refined Correspondence Estimation

After obtaining dense depth maps for any image A, we reestimate correspondences once with updated geometric constraints $\tilde{\mathbf{w}}_{AB}$. During this second iteration of correspondence estimation, we reevaluate the weighting term $\tilde{\alpha}$ based on the confidence in the depth reconstruction: $\tilde{\alpha} = \alpha \cdot \max(e_{max} - e_{repr}(\mathbf{x}), 0)/e_{max}$, typically $e_{max} = 10px$. It only affects the optimization if $e_{repr}(\mathbf{x}) < e_{max}$. Again, the reason for the introduction of the geometric constraint is to steer the correspondence search towards geometrically plausible solutions. Where the initially computed correspondence maps result in a plausible geometric proxy (i.e., $e_{repr} < e_{max}$), the correspondence search is heavily influenced by the depth information. In regions where they do not agree on a valid model, the correspondence search remains unconstrained. Effectively, this scheme constitutes a hybrid reconstruction: whenever plausible geometric information can be derived, the correspondence search is forced to be consistent with it. Otherwise, we accept a geometrically unconstrained solution.

By employing both a symmetric (12) and a geometric term (13) simultaneously, we encourage consistency across the whole data set. For scenes with large depth discontinuities, we also allow to incorporate a segmentation of the foreground object(s). The two different layers are processed independently to prevent mismatches between fore- and background. Matching costs between pixels of different layers are set to same default (high) value that also applies to potential out-of-image correspondences, as proposed by Liu et al. [24].

VI. Rendering

We use a GPU-based multi-image warping and blending scheme. As input data, we have a set of (calibrated) images as well as a weighting function that determines for every virtual view V a subset $S = \{A_1, A_2, ..., A_n\}$ of all images and a set of weights $T = \{a_1, a_2, ..., a_n\}$, $|S| = |T| = n$. As a weighting scheme we use the per-camera weighting of Pulli et al. [28], depending on the camera setup, we set $n \in 2, 3, 4$. Images A_i, A_j that are contained in at least one common subset S are referred to as neighbors. We also make use of the depth maps D_{A_i} and the correspondence maps $\mathbf{w}_{A_iA_j}$ between neighboring images.

We compute world space correspondence maps $\mathbf{W}_{A_iA_j} = X_{A_j} - X_{A_i} = C_{A_j}^{-1}(\mathbf{x} + \mathbf{w}_{A_iA_j}(\mathbf{x})) - C_{A_i}^{-1}(\mathbf{x})$. World space correspondence maps $\mathbf{W}_{A_iA_j}$ are only considered valid in regions where $\mathbf{w}_{A_iA_j}$ is symmetric. We propagate world

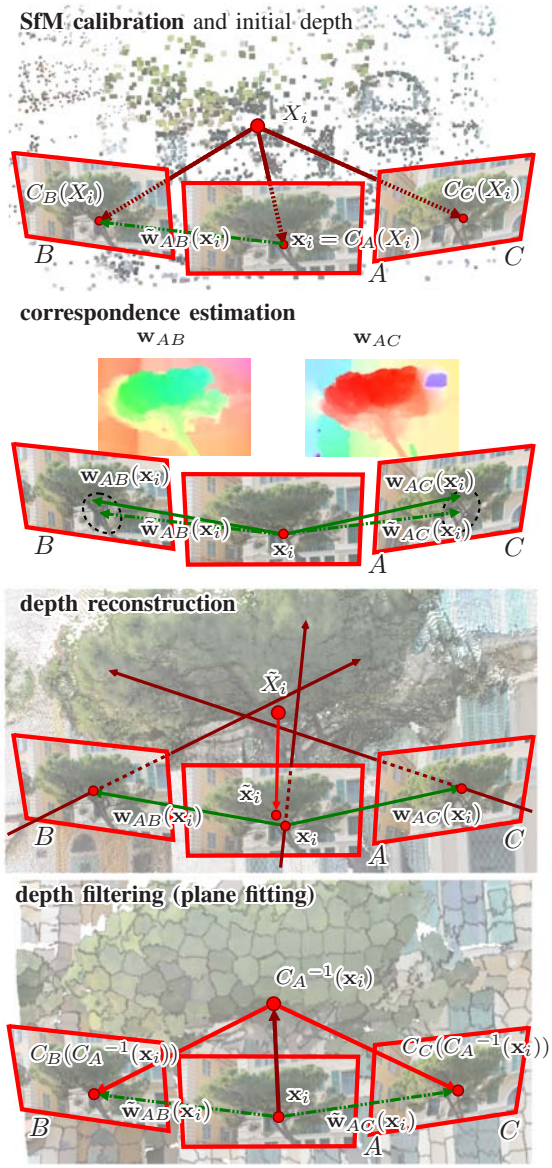

Fig. 3: Reconstruction of Tree18. First, we use **SfM** to calibrate the input images. We reproject single points X_i of the sparse 3D point cloud to the image planes of A, B, C. **Correspondence estimation** between neighboring images is performed. We enforce correspondences $\mathbf{w}_{AB}$ to be similar to the initial correspondences $\tilde{\mathbf{w}}_{AB}$ derived from the geometric model. **Depth reconstruction** is based on triangulation of corresponding points. We approximate $\tilde{X}_i$ as the point with the minimal squared distance to all rays. The reprojection error $e_{repr} = |\tilde{\mathbf{x}}_i - \mathbf{x}_i|_2$ indicates if the reconstructed depth is valid. **Depth filtering** is applied via plane fitting on each 2D (SLIC) superpixel. The new depth values are used to update the geometric constraints $\tilde{\mathbf{w}}_{AB}$, which are used for a second iteration of correspondence estimation.

space correspondences to invalid regions using anisotropic diffusion [27] with two modifications: First, only regions with invalid values are updated in each iteration. Second, the depth value $D_A(\mathbf{x})$ is used for the evaluation of the nearest-neighbor difference. This ensures propagation only on spatially connected objects.

We render a dense grid of quads, where every vertex represents a pixel location $\mathbf{x}$ in image A_i. Using the depth information for A_i, we calculate the world space coordinate X for a given vertex $\mathbf{x}$. According to $\mathbf{W}_{A_iA_j}$, we warp it towards its corresponding world space position in A_j by weight a_j. This warping step is repeated for every image $A_k \in S$ with weight a_k. We reproject the final world space location onto the image plane of V. In order to handle disocclusions, e.g. areas where two objects seem to move apart, we break up our mesh when two neighboring vertices $\mathbf{x}_i, \mathbf{x}_k$ differ too much in their depth values:

$max(\mathbf{x}_k.d/\mathbf{x}_l.d, \mathbf{x}_l.d/\mathbf{x}_k.d) > e_{depth}$, typically $e_{depth} = 1.1$.

We repeat this procedure for all images $A_j \in S, A_j \neq A_i$ and obtain n different projections. When blending the n different reprojections, we use a soft z-Buffer as well as inpainting technique as described by Zheng et al. [36].

VII. Results and Evaluation

In order to give an understanding of the reconstruction and rendering workflow, we present and discuss results for five test scenes and highlight the separate challenges we encountered, cf. Fig. 4. We present free-viewpoint renderings for all scenes and compare our approach to the state of the art in free-viewpoint rendering in Sect. VII-A. We further conduct a quantitative evaluation using the Naturalness Image Quality Evaluator (NIQE) in Sect. VII-B.

A. Qualitative Evaluation

To assess the visual quality of our approach, we would like to refer to the video submitted along with our paper. Four of the scenes are publicly available data sets that have been used by other state-of-the-art free-viewpoint algorithms [2], [8], [22]. For our comparison, we deliberately chose pairings of algorithms and scenes that provide average- to best-case results, as reported in the individual publications [2], [8], [22]. By directly comparing against these very different state-of-the-art algorithms we would like to highlight the generality of our approach. Please note that all scenes show dynamic content: None of the data sets feature synchronous image captures, they have either been recorded by a single moving DSLR camera or several non-genlocked camcorders. All scenes feature image regions that do violate the epipolar constraint to some extent. This manifests in a high reprojection error of non-static objects, i.e. treetops (Tree18, Clouds), clouds (Clouds) and moving actors (Juggler, Freeclimbing, WhoCares).

Due to the diversity of the test scenes, a full evaluation of all state-of-the-art algorithms on all test scenes is infeasible: Unstructured VBR [2] is designed to work well on static scenes with a single moving actor. The WhoCares, the Tree18 and the Clouds scene have differents setup (multiple actors, no foreground object and dynamic background, respectively) and would not provide pleasing results. For the Freeclimbing sequence, we reimplemented their rendering algorithm (DIBR+billboards) to give an estimate of the expected visual quality. The Silhouette-Aware Warping [8] can cope with

Fig. 4: Comparison of state of the art (left) and our hybrid approach (right), top to bottom: **Tree18**: In contrast to conventional DIBR [8], our hybrid approach aligns the semi-transparant car windows. **Clouds**: Panorama stitching [3] may display ghosting artifacts for moving scene contents. Our approach successfully aligns the clouds. **Juggler**: Cross-blending of billboards may result in ghosting artifacts [2], our piecewise planar depth and warping-based alignment remedy this effect. **Freeclimbing**: We compare a blend of two consecutive frames of a billboard-based render to our hybrid approach. Despite the wide baseline, the freeclimber can be aligned. **WhoCares**: For the production of the WhoCares music video, manual retouching of correspondences and rendered frames has been employed [22], [23]. Our approach achieves comparable quality without user interaction.

scenes where depth reconstruction fails in certain areas. Since large parts of the Freeclimbing, WhoCares, Juggler and Clouds scene violate assumption made in depth reconstruction, the manual labor spent for depth correction would certainly be very high. The quality of the rendered video would rather reflect the skill and stamina of the human operator than the power of the underlying free-viewpoint system. The same reasoning applies to the Virtual Video Camera [22] that relies on manual correspondence correction in challenging scenes.

Tree18 dataset: We used all 18 photographs of the Tree18 dataset provided by [8], they were downsized to 1296×864 px before further processing. Due to the arc-like setup, we set $n = 2$, cf. Sect. VI, so that the two closest original views are used to render an in-between view. The sparse depth information [14] was used to constrain the initial correspondence map estimation. In our video comparison, we show that our approach can compete with the state of the art [8]. While our rendering was created without any additional input, Chaurasia et al. [8] allow the user to give cues for edges and depth. The moving treetop and the semi-reflective car windows violate basic assumptions of DIBR and depth is challenging to reconstruct, as reported by Chaurasia et al. [8]. Since the (unconstrained) correspondence search yields plausible results for these images regions, our approach can produce visually pleasing renderings, cf. Fig. 4.

Cloud dataset: Images were captured with a DSLR and processed at 1620×1080 px. The scene was segmented into fore- and background with Nuke rotoscoping tools [34]. Both layers were processed and rendered independently. We took 140 photos of the scene, half of them were taken in a standing and the other half in a crouching position. They span a full circle around the statue in the center. We set $n = 4$, so that two images from the upper row and two images from the lower row contribute to a virtual view. Although the scene seems easy to process due to the high density of views, the dynamic sky in this scene poses a hard challenge: Since the clouds have been moving during the ten minute capturing process, no valid depth has been reconstructed. Zimmer et al. [37] reported that only an unconstrained optical flow is able to cope with this kind of scene elements. Using our unconstrained correspondence search for these areas, we are able to align the clouds in the final render.

Juggler dataset: Six handheld camcorders (960×544 px) have been used to capture this sequence. We used the Unstructured Video Based Rendering application by Ballan et al. [2] to create a short time-freeze free-viewpoint video clip. We recreated the camera motion with our hybrid approach. Similar to the Tree18 scene, we set $n = 2$. As input, we used the publicly available segmentation data and background reconstruction. In contrast to the billboard-switching approach of Ballan et al., our hybrid approach provides a consistently deforming actor, cf. Fig. 4. While their interactive viewer focusses on transitions between two neighboring cameras, we can render a single continuous camera arc. We would like to refer to our video for a direct comparison.

Freeclimbing dataset: The biggest challenge on this data set was the wide camera baseline. The four camcorders have been spaced $> 40\,^\circ$ apart. Due to the rhomb-shaped setup, we

set $n = 3$, so that three of the four original images contribute to the final synthetic render. Ballan et al. [2] used a foreground segmentation and a billboard approximation of the freeclimber for free-viewpoint rendering. Similar to their approach, we obtained a foreground segmentation and processed both layers independently. This enabled setting different parameters to the foreground, i.e. we set $e_{max} = 20px$, otherwise, no valid geometry can be obtained. We used four images (1920×1080 px) for the free-viewpoint transition shown in the video. This also allows to render arbitrary camera-paths among the whole set of cameras. In order to assess the contribution of our approach, we produced three renders of this scene. First, we used a single fronto-parallel billboard for the actor, cf. Fig. 4 (left). Second, we used our reconstruction pipeline to obtain a per-pixel depth value (cf. submission video). Third, we used our full approach of hybrid reconstruction and rendering, cf. Fig. 4 (right).

WhoCares dataset: Since this scene has been captured in a green-screen environment, chroma-keying allowed an unsupervised foreground segmentation. Eleven unsynchronized camcorders (1920×1080 px) were used for the capture, we processed several dozen images of each camera. Lipski et al. [22] used their rendering pipeline to create a free-viewpoint music video. We recreated a five-second sequence of their music video with our approach. Since they move the camera in an arc along the original camera positions, we set $n = 2$. In order to achieve convincing results, they relied on elaborate tools for correspondence map correction [19], manual retouching with GIMP and manual compositing of the different layers. For comparable quality, our hybrid approach did not require any manual retouching of the rendered data, cf. Fig. 4.

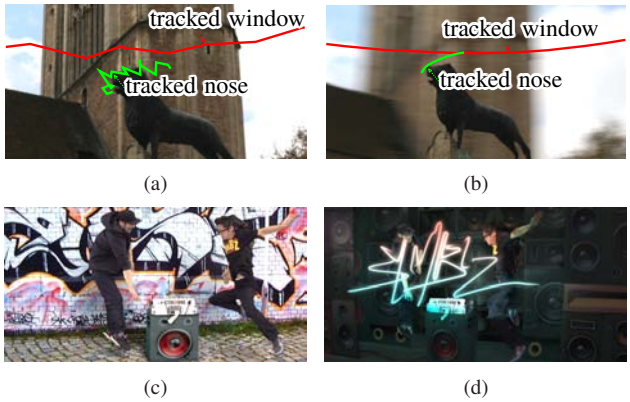

Fig. 5: Further Applications. We rerendered the cloud data set (a) with novel views placed on a perfect ellipse (b). By pointing the viewing rays to the scene center and synthesizing motion blur, we created a "super-stabilized" virtual camera. This is visualized by tracking a church window in the background and the nose of the foreground statue across multiple frames in (a,b). Our free-viewpoint rendering can be composited with matchmoved camcorder footage (c). We also exploit the geometric reconstruction to relight the scene (d).

B. Quantitative Evaluation (NIQE)

In order to properly evaluate the quality of our rendering algorithm, we also conduct a quantitative analysis. Full-reference metrics, e.g. Peak Signal to Noise Ratio (PSNR) give an image quality estimate by pixel-wise comparing a rendered image to its pristine counterpart. Given the unique properties of image-based free-viewpoint rendering, this methodology is quite problematic in our case. If we try to re-render one of the source images with our method, the resulting synthetic image would simply be an exact copy of the source image. This also applies to other state-of-the-art algorithms [2], [8], [22]. Another option would be to remove one of the original images from the set of available source images and try to recreate it (leave-one-out tests). Since the cameras are not synchronized, the in-between view cannot be exactly reproduced: Even if a visually plausible interpolation can be achieved with our or other algorithms, a pixel-wise comparison would yield unsatisfactory results.

Fortunately, recent advances in automatic quality assessment make it possible to conduct an quantitative evaluation of the perceived image quality. We make use of the Natural Image Quality Evaluator (NIQE) [26]. Instead of a per-pixel comparison to reference images, a quality score is obtained by analyzing image properties with respect to learned image statistics. We randomly divide the input images into two subsets of identical size. The first subset and is used to train NIQE. The second is used to obtain a reference score, cf. Table I(left). Since all data sets contain some artifacts caused by the capturing system (e.g., blur, noise or compression), the reference score is used as an estimate for the best-case results of any image synthesis. We compute scores for our approach and state-of-the art algorithms, cf. Table I.

	reference		our method		state-of-the-art		
	NIQE score	no. imgs	NIQE score	no. imgs	method	NIQE score	no. imgs
Tree18	2.421	9	3.189	151	SA.Warp. [8]	3.470	151
Juggler	3.418	12	4.827	151	Unstr.VBR [2]	5.374	151
Freeclimbing	5.445	4	7.884	40	DIBR+bb	7.581	40
WhoCares	9.075	11	10.654	155	VVC [22]	17.36	155
Clouds	3.196	70	4.171	70			

TABLE I: Quantitative comparison to the state of the art. Our approach achieves comparable or better results (=low scores) than other algorithms.

For the Tree18, the Juggler and the WhoCares scene, we outperform other approaches. Only in the Freeclimbing scene, DIBR+billboards achieves slightly better (=lower) scores. We would like to point out that DIBR+billboards exhibits artifacts like popping and floating that are not possibly captured by any single-image quality estimator.

C. Limitations

Additional input data helps to increase the robustness and the quality of our approach. For three of the five shown scenes (Freeclimbing, Clouds, Juggler), an additional manual foreground segmentation helped to obtain valid correspondences and depth values. We argue that image segmentation is a more common and simple task compared to depth [8] or correspondence correction [19]. Still, a fully automatic processing would be preferable. A similar problem is the

automatic SLIC super-pixel segmentation of the input images. Unlike correspondences or depth, the initial segmentation is not refined during our iterative scheme. This results in spurious rendering artifacts, e.g., popping. Promising research has been conducted that aims to find a global solution to depth and segmentation simultaneously [5] and could be incorporated into our approach.

Although we can cope with a sparse camera placement as shown in the Freeclimbing scene, our approach is not feasible for all setups. E.g., the Rothman scene from [2] features cameras that are too widely spaced apart to allow for a faithful reconstruction using image correspondences. In the most extreme case (Freeclimbing scene) we were able to bridge distances of 40°, so one would need at least nine cameras to render a full 360° path.

Another drawback are the relatively long processing times for each scene. Similar to [22], pairwise correspondence maps have to be estimated between neighboring images. Depending on image resolution and possible correspondence vector magnitudes, computing pairwise correspondences takes minutes or hours per image pair. In addition, depth estimation and a second iteration of correspondence estimation lead to processing times that are at least twice as long as the state of the art [22]. The actual rendering process only takes a few dozen milliseconds, with data i/o as the limiting factor.

VIII. Applications

Instead of devising our own free-viewpoint editing framework, we use our renderer as a backend for existing tools. After the reconstruction phase, cf. Sect. V, we export a simple 3D proxy geometry of the scene along with the camera positions. We devised scene importers to compositing and 3D modeling software (Nuke and Blender). Camera paths can be created using standard animation tools. The final camera path is exported to our renderer. Different applications are conceivable with our hybrid approach that are difficult or impossible to recreate with other techniques:

Image Stabilization: We rendered a "super-stabilized" camera trajectory for the clouds data set. We placed the virtual camera on a perfect ellipse and rerendered the image sequence with the virtual camera constantly pointing towards the central statue. In order to visualize the steady movement of objects in image space, we plotted the tracking results for two objects in the scene, cf. Fig. 5 (a,b). We simulated a 1/50s shutter by averaging multiple renderings along this trajectory. By using purely image warping based techniques, this kind of view extrapolation is not possible.

Panorama Creation: Since we achieved a believable 360 ° rendering of the background, we can easily create a panoramic representation of the scene by combining small (3-pixel wide) slices of the output images. Panorama stitching algorithms often assume that both camera and scene content remain static during acquisition [3]. Therefore, the rendered panorama often displays ghosting artifacts when these assumptions are violated, cf. Fig. 4.

Compositing: We augment the camcorder-recording of a graffiti-covered brick wall with a free-viewpoint rendering of the WhoCares foreground layer. The movement of the hand-held camera is tracked using matchmoving software (Camera-atracker [34]). Additionally, 3D objects and contact shadows are inserted into the scene, cf. Fig. 5 (c). For compositing of the different layers, we rendered the contents of the z-Buffer and used the depth data for a deep compositing in Nuke.

Re-lighting: In a 3D modeling software, the captured scene can be aligned with 3D objects. The proxy geometry is accurate enough for rendering soft shadows and receiving lighting from the 3D environment. The render of the virtual view along with shadows and lighting data are used for a final composite. We showcase these possibilities in our submission video. A neon light source is inserted into the scene and the proxy geometry receives lighting information. During compositing, the diffusely lit geometry is merged with the actual free-viewpoint rendering to create the illusion of a neon light source, cf. Fig. 5 (d).

IX. Conclusion

We present a versatile and robust hybrid formulation for free-viewpoint rendering. Our approach can compensate for inaccurately estimated scene geometry by incorporating visually plausible correspondences into the rendering equation. We link depth and correspondence estimation by a soft geometric constraint in our iterative reconstruction pipeline. We demonstrate the versatility of our approach on a wide variety of test scenes and applications. We experimentally compare our results with the state of the art, showing that our hybrid rendering scheme works without any additional user input (Tree18, WhoCares) or produces better results (Clouds, Juggler).

Acknowledgements

We would like to thank the authors of [2], [4], [8], [22] for sharing their multi-view data sets and the authors of [2], [3] for providing implementations of their algorithms. We would also like to thank The Foundry Ltd. for providing Nuke software licenses. This was funded by the European Union's Seventh Framework Programme FP7/2007-2013 (grant no. 256941, "Reality CG") and by the DFG (MA 2555/1-3, 2555/8-1).

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurlien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels. Technical report, EPFL, 2010.

[2] Luca Ballan, Gabriel J. Brostow, Jens Puwein, and Marc Pollefeys. Unstructured Video-Based Rendering: Interactive Exploration of Casually Captured Videos. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 29(3):87:1–87:11, July 2010.

[3] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, August 2007.

[4] Thomas Brox, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers. Combined Region and Motion-Based 3D Tracking of Rigid and Articulated Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:402–415, 2010.

[5] Neill D.F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Automatic object segmentation from calibrated images. In *Visual Media Production (CVMP), 2011 Conference for*, pages 126 – 137, Nov. 2011.

[6] Thomas J. Cashman and Andrew W. Fitzgibbon. What Shape are Dolphins? Building 3D Morphable Models from 2D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2012.

[7] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics*, 32(3):30:1–30:12, June 2013.

[8] Gaurav Chaurasia, Olga Sorkine, and George Drettakis. Silhouette-Aware Warping for Image-Based Rendering. *Computer Graphics Forum*, 30(4):1223–1232, 2011.

[9] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proc. of ACM SIGGRAPH'93*, pages 279–288, New York, 1993. ACM Press/ACM SIGGRAPH.

[10] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, August 2008.

[11] Per Einarsson, Charles-Felix Chabert, Andrew Jones, Wan-Chun Ma, Bruce Lamond, Tim Hawkins, Mark Bolas, Sebastian Sylwan, and Paul Debevec. Relighting Human Locomotion with Flowed Reflectance Fields. In *EGSR 2006*, pages 183–194, June 2006.

[12] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating Textures. *Computer Graphics Forum (Proc. Eurographics EG'08)*, 27(2):409–418, 4 2008.

[13] Christoph Fehn. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. *Proc. SPIE*, 5291(93):1–1, 2004.

[14] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, August 2010.

[15] Marcel Germann, Alexander Hornung, Richard Keiser, Remo Ziegler, Stephan Würmlin, and Markus Gross. Articulated Billboards for Video-based Rendering. *Comput. Graphics Forum (Proc. Eurographics)*, 29(2):585–594, 2010.

[16] Marcel Germann, Tiberiu Popa, Richard Keiser, Remo Ziegler, and Markus H. Gross. Novel-view synthesis of outdoor sport events using an adaptive view-dependent geometry. *Comput. Graph. Forum*, 31(2):325–333, 2012.

[17] Michael Goesele, Jens Ackermann, Simon Fuhrmann, Carsten Haubold, Ronny Klowsky, and TU Darmstadt. Ambient point clouds for view interpolation. *ACM Trans. Graph.*, 29:95:1–95:6, July 2010.

[18] Jean-Yves Guillemaut and Adrian Hilton. Joint Multi-Layer Segmentation and Reconstruction for Free-Viewpoint Video Applications. *International Journal of Computer Vision*, 93:73–100, 2011. 10.1007/s11263-010-0413-z.

[19] Felix Klose, Kai Ruhl, Christian Lipski, and Marcus Magnor. Flowlab - an interactive tool for editing dense image correspondences. In *Proc. European Conference on Visual Media Production (CVMP) 2011*, 2011.

[20] Manuel Lang, Alexander Hornung, Oliver Wang, Steven Poulakos, Aljoscha Smolic, and Markus Gross. Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. Graph.*, 29(3):10, 2010.

[21] C. Lipski, C. Linz, T. Neumann, M. Wacker, and M. Magnor. High Resolution Image Correspondences for Video Post-Production. In *CVMP 2010*, volume 7, pages 33–39, Los Alamitos, CA, USA, November 2010. IEEE Computer Society.

[22] Christian Lipski, Felix Klose, Kai Ruhl, and Marcus Magnor. Making of Who Cares? HD Stereoscopic Free Viewpoint Video. In *Proc. European Conference on Visual Media Production (CVMP) 2011*, volume 8, pages 1–10, November 2011.

[23] Christian Lipski, Christian Linz, Kai Berger, Anita Sellent, and Marcus Magnor. Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time. *Computer Graphics Forum*, 29(8):2555–2568, 2010.

[24] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. SIFT Flow: Dense Correspondence across Different Scenes. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 28–42, 2008.

[25] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[26] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013.

[27] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[28] Kari Pulli, Michael Cohen, Tom Duchamp, Hugues Hoppe, Linda Shapiro, and Werner Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In *In Eurographics Rendering Workshop*, pages 23–34, 1997.

[29] Kai Ruhl, Felix Klose, Christian Lipski, and Marcus Magnor. Integrating approximate depth data into dense image correspondence estimation. In *Proc. European Conference on Visual Media Production (CVMP) 2012*, August 2012.

[30] Aljoscha Smolic. Computer Analysis of Images and Patterns, 13th International Conference, CAIP 2009, Münster, Germany, September 2-4, 2009. Proceedings. 5702, 2009.

[31] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. on Graphics*, 25(3):835–846, 2006.

[32] Jonathan Starck, Atsuto Maki, Shohei Nobuhara, Adrian Hilton, and Takashi Matsuyama. The multiple-camera 3-d production studio. *IEEE Trans. Cir. and Sys. for Video Technol.*, 19(6):856–869, June 2009.

[33] Timo Stich, Christian Linz, Christian Wallraven, Douglas Cunningham, and Marcus Magnor. Perception-motivated interpolation of image sequences. *ACM Transactions on Applied Perception (TAP)*, 8(2):1–25, January 2011. http://doi.acm.org/10.1145/1870076.1870079.

[34] The Foundry. Nuke/Cameratracker website. http://www.thefoundry.co.uk/products/, May 2012.

[35] G. Xú and Z. Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996.

[36] Ke Colin Zheng, Alex Colburn, Aseem Agarwala, Maneesh Agrawala, David Salesin, Brian Curless, and Michael F. Cohen. Parallax photography: creating 3d cinematic effects from stills. In *Proceedings of Graphics Interface 2009*, GI '09, pages 111–118, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

[37] H. Zimmer, A. Bruhn, and J. Weickert. Freehand HDR imaging of moving scenes with simultaneous resolution enhancement. *Computer Graphics Forum (Proceedings of Eurographics)*, 30(2):405–414, 2011. Web page: http://www.mia.uni-saarland.de/Research/SR-HDR/.

[38] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon A. J. Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004.

Christian Lipski studied computer science at University of Manchester, UK, and TU Braunschweig, Germany. He received his Diploma degree in Computer Science from TU Braunschweig in 2006 and his PhD in 2013. He was awarded with the BBC Best Paper award 2010 and the SAE alumni award in 2012. His research interests include stereoscopic and free-viewpoint video, visual effects in digital cinematography and image based rendering.

Felix Klose received a Diploma degree in Computer Science from TU Braunschweig in 2010. He is currently pursuing his PhD in Computer Graphics at TU Braunschweig, Germany. His main research interests include depth and scene flow reconstruction as well as stereoscopic and free-viewpoint rendering.

Marcus Magnor is full professor and head of the Computer Graphics Lab at TU Braunschweig. He holds a Diploma degree in Physics (1997) and a PhD in Electrical Engineering (2000). After his post-graduate time in the Graphics Lab at Stanford University, he established his own research group at the Max-Planck-Institut Informatik in Saarbrücken. He completed his habilitation and received the venia legendi in Computer Science from Saarland University in 2005. In 2009, he spent one semester as Fulbright scholar and Visiting Associate Professor at the University of New Mexico. His research interests meander along the visual information processing pipeline, from image formation, acquisition, and analysis to image synthesis, display, perception, and cognition.