TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/09 pp678-695
Volume 21, Number 6, December 2016

A Survey on Multiview Video Synthesis and Editing

Shaoping Lu, Taijiang Mu*, and Songhai Zhang

Abstract: Multiview video can provide more immersive perception than traditional single 2-D video. It enables both interactive free navigation applications as well as high-end autostereoscopic displays on which multiple users can perceive genuine 3-D content without glasses. The multiview format also comprises much more visual information than classical 2-D or stereo 3-D content, which makes it possible to perform various interesting editing operations both on pixel-level and object-level. This survey provides a comprehensive review of existing multiview video synthesis and editing algorithms and applications. For each topic, the related technologies in classical 2-D image and video processing are reviewed. We then continue to the discussion of recent advanced techniques for multiview video *virtual view synthesis* and various *interactive editing applications*. Due to the ongoing progress on multiview video synthesis and editing, we can foresee more and more immersive 3-D video applications will appear in the future.

Key words: multiview video; view synthesis; video editing; color correction; survey

1 Introduction

The increasing availability and diminishing prices of various video cameras, depth sensors, and multi-camera systems have caused an enormous growth of 3-D-oriented video applications. It also opens a variety of new opportunities in 3-D Television (3-DTV)[1], Free-view Television (FTV)[2], and many other domains.

Multiview videos, which are recorded from different viewpoints by multiple synchronized cameras, can be visualized on creative 3-D immersive displays. An example is autostereoscopic display which enables different viewers to perceive motion parallax and experience free viewpoint video. Intuitively, arranging denser cameras can capture more video streams from different discrete viewpoints, and those abundant multiview videos could better support high-quality realistic 3-D displays. However, this imposes an enormous burden on the acquisition, storage, compression, and transmission of multiview video data.

Efficiently displaying realistic 3-D scenes based on multiview videos that are captured from limited viewpoints at a high level of quality is still a long way from reality. Accurate 3-D modeling-based multiview techniques are still impractical, since the quality of fully automatic 3-D reconstruction usually is not sufficient and extremely tedious user interactions, even for professionals, are required in the non- or semi-automatic methods. With the advent of various advanced technologies on image and video synthesis and editing, multiview video based research and applications have attracted increasing attention, and gave rise to the well-studied Depth Image-Based Rendering (DIBR) techniques[3, 4], which attempt to synthesize many additional viewpoints based on a limited set of given views.

A typical multiview video processing framework (see Fig. 1) can be generally separated into the following different phases: (1) data acquisition, (2) multiview representation, (3) compression and transmission, (4) rendering, and (5) display processing. In the multiview

• Shaoping Lu is with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel (VUB), B-1050 Brussels, Belgium. E-mail: slu@etrovub.be.
• Taijiang Mu and Songhai Zhang are with the TNList, Tsinghua University, Beijing 100084, China. E-mail: mmmutj@gmail.com; shz@tsinghua.edu.cn.
∗ To whom correspondence should be addressed.
Manuscript received: 2016-10-14; accepted: 2016-10-21

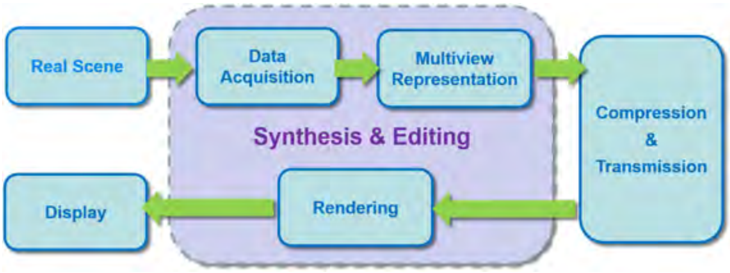

Fig. 1 A typical framework of the multiview video system. In this kind of system, video synthesis and editing can be performed in data acquisition, multiview representation or rendering phases.

video acquisition step, it is crucial that all cameras are accurately calibrated and synchronized. It also needs to be ensured that all cameras have the same color balance. To support free viewpoint navigation and various creative editing applications, synthesizing the scene content of desired views is of high importance. Therefore, this survey mainly focuses on *color correction*, *view synthesis*, and *interactive editing* with multiview-plus-depth (MV+D) video as the main representation format. Other potential 3-D representation formats[5] include meshes, point clouds, patch clouds, volumetric models or layered models. However, the MV+D format is often the most convenient as it is the most closely related to the way the data was acquired.

Essentially, multiview video processing methods analyze and model the visual data captured by multiple cameras, and most of the relevant techniques originated from single image/video processing research. Hence, for each topic we also briefly review the related progress of single image/video processing. Note that there are many other challenges that are related to the synthesis and editing problem but fall outside the scope of this survey. Examples are accurate depth estimation[6], efficient compression[7], and transmission[8] of the massive multiview data.

The organization of this paper is as follows. Section 2 discusses the advances on multiview color correction. Section 3 classifies existing multiview video synthesis approaches. Section 4 discusses various applications of multiview video editing. Finally, the conclusion with discussions of the future research directions is summarized in Section 5.

2 Multiview Color Correction

The color that is measured by visual sensors does not only rely on the light source, but also on the geometric surface and the appearance properties of the 3-D object. Although in the literature variable illumination based color transformation and evaluation models exist (e.g., Ref. [9]), for the sake of computational complexity, most of the existing multiview color correction methods focus on transforming all colors of the input image to those as they appear in a reference view under the same light source. Basically, this can be seen as a specific recolorization processing. In this context, we will focus on the techniques of colorization-based interaction and multiview color correction processing.

2.1 Image color interaction

Color correction related image/video color processing and interaction has attracted widespread interest in many research domains, e.g., image recolorization, multi-scale texture and exposure adjustment, appearance editing propagation, correspondence matching, style transfer, as well as content alignment between images.

One of the pioneering works in color editing was introduced by Reinhard et al.[10], where the target color is automatically adapted according to a reference image. This method efficiently scales the high-frequency components of the input textures using the respective color standard deviations of the target and reference images; however, this may result in over-saturated colors in the target images. In Ref. [11], the colorization is firstly modeled as an optimization problem, and the authors employed a multigrid solver to perform the desired color for video. Lischinski et al.[12] further observed that this sparse interpolation-like colorization can be applied for exposure and other tonal adjustment. A similar strategy is introduced for multi-scale texture decomposition[13]. In this multi-resolution framework, the texture can be easily enhanced by increasing the high-frequency details when reconstructing the image. Inspired by this idea, Xiao and Ma[14] proposed to highlight the gradient information in the color transfer work. In order to avoid color bleeding artifacts between areas of different colors, the local gradient-saliency[15] was introduced into an interactive colorization optimization framework, with which the relatively important content in some image areas can be better handled in various color processing applications.

How to effectively and efficiently perform color editing is an attractive research topic. In order to efficiently propagate the user-assigned color (and the appearance) to the entire image, in Ref. [16], a low-rank stochastic approximation method is introduced to solve a sparse linear system. Xu et al.[17] constructed a KD tree in the high-dimensional feature space to perform the acceleration before user interaction. However, building the KD tree is still expensive in terms of both runtime and memory. Thus, Li et al.[18] used Radial Basis Functions (RBF) to interpolate the propagation method, by which the user can instantly get the feedback of color interaction (see Fig. 2). After that, Chen et al.[19] employed Locally Linear Embedding (LLE)[20] to accurately represent the geometric structure and to propagate edit operations specified by a user. Following this idea, in Ref. [21] an adaptive pixel neighborhoods decision model is introduced to improve the representation of each pixel's manifold structure. By jointly considering spatial distance, sample location, and appearance, a sparse editing model[22] is proposed to intelligently propagate the desired color on object level, where a high-dimensional Gaussian filtering is employed and thus much less color samples are needed in comparison with other methods. Interestingly, due to the excellent advantages of latest neural networks on feature classification and learning, deep learning based colorization is also being explored. For example, to perform gray-to-color conversion in Ref. [23], the target chrominance of a gray image is obtained from a reference database with a neural network where the connections between neurons are based on extracted pixel-level feature descriptors. This work heavily depends on high-quality segmentation of an image, and the network would introduce noise around low-texture areas. To remove such noise, the authors further applied a joint bilateral filter to smooth the learned color. In comparison, another method proposed in Ref. [24] jointly learns global and local features for an image, and it works in an end-to-end style from a large dataset to generalize to various types of images. Note that these methods are imperfect for semantic-level colorization due to the limitation of robust scene understanding.

Color correction is also well-studied in many image composition-oriented applications. Taking image stitching as an example, Afifi and Hussain[25] proposed a modified poisson blending technique to reduce the color bleeding artifacts by leveraging pixels from both the source and the target images boundaries in the blending process. Qian et al.[26] performed manifold alignment not only to preserve the local geometries of color distribution but also to match corresponding pixels. Besides that, color correction is applied to align the temporal appearance fluctuation for photo collections[27, 28] and videos[29]. Similarly, a color state smoothing processing is proposed with a ℓ_1 optimization model for video tonal adjustment in Ref. [30], where a color state is a representation of the exposure and white balance of a frame. The blind temporal consistency[31] is then introduced to automatically change the video rendering style. This method considers the temporal consistency and scene dynamics using time-varying scene warping under a general optimization model, so it is still sensitive to the correspondence construction. Researchers also considered aligning the color for video sequence matching (see an example in Ref. [32]). Recently, color retargeting[33] has been introduced for the user to

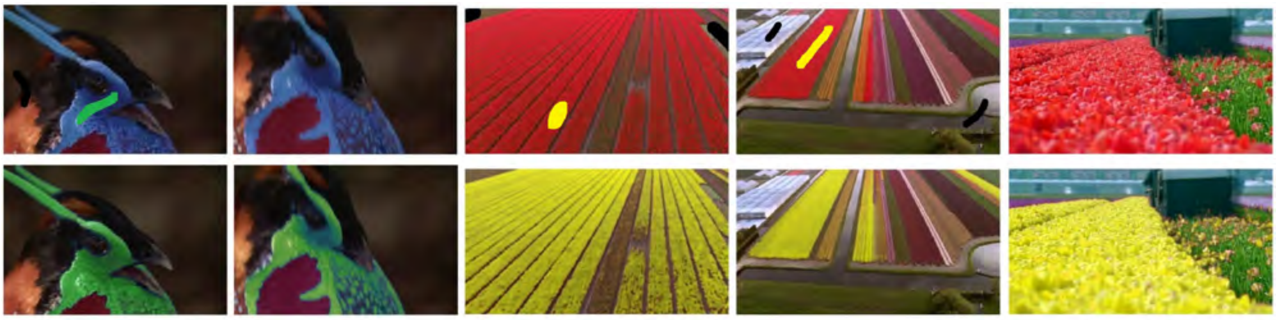

Fig. 2 Efficient color propagation for images and videos. Examples are from Refs. [17, 18].

interactively edit the variable color. In such method, the output image is efficiently generated by optimally resampling the pixels from multiple time-lapse images. In general, by benefiting from the ongoing advances on various image color processing techniques, future image/video color applications are being facilitated towards real-time manipulation and more intelligent interaction.

2.2 Multiview color correction

Because multiview video is recorded by different cameras, apparent inter-view color variations would be generated due to uncalibrated camera parameters, global lighting conditions, etc. Misaligned colors would result in visual fatigue, binocular rivalry, and other negative 3-D viewing effects. Moreover, inter-camera color inconsistencies may severely affect the view synthesis quality and multiview compression performance. Thus, multiview color correction is usually applied with a color mapping, by which the color of the input view is adjusted to become as consistently as possible to that of the reference view. In comparison with single image based color processing techniques, multiview color correction further exploits both temporal consistencies in the same viewpoint and inherent coherency between multiple viewpoints.

In a typical multiview video processing pipeline (see Fig. 1), color correction can be applied as a part of the data acquisition unit[34]. However, most of existing solutions are either (1) carried out as a prefilter before compression or representation, (2) integrated inside the video encoder, or (3) post-processed after the video stream is decoded.

As a prefilter. Examples in this subclass include Refs. [35–47]. To better support the 3-D reconstruction performance, the system proposed in Ref. [36] comprises camera calibration and software-based color correction as a two-phase iteration, as frequent calibration is usually time-consuming or even impractical. In Refs. [37, 38] temporal histogram matching based color correction is performed, and the authors demonstrated that their work greatly improves the compression efficiency. Similarly, many other color prefilting methods are introduced with the goal of lowering bit-rates for transmission. The methods proposed in Refs. [39, 40] further consider to use block-based matching, which is also a commonly employed processing block in the encoder. In Ref. [43], color correction is first applied on specific keyframes. A temporal variations model, which is constructed to detect time-invariant regions, is then used to adjust correction coefficients for other non-keyframes. In Ref. [45] the histogram matching is built on a Group Of Pictures (GOP). A 3-D lookup table (corresponding to 3 color channels) implemented on GPU is also introduced in Ref. [42] for fast color correction. As demonstrated in Refs. [37, 39–41, 43–46], when all videos are well aligned using color correction by such prefilters, the efficiency of both inter-view prediction as well as motion-compensated prediction is increased, which in turn results in improved multiview video compression performance. On the other hand, some prefilters would also introduce negative visual artifacts or even generate over-smoothing results. Hence the authors in Refs. [35, 47] suggested that multiview color correction should pay more attention on preserving the structure information of the original videos, since the well preserved textures are critical for content-aware rendering (e.g., virtual view synthesis) and further applications of interactive editing.

Inside the encoder. Color correction integrated into the video encoder aims at better reusing the matching, motion vectors, and other information provided by the video encoder. Hence, integrating the color correction inside the encoder can avoid some redundant computations such as block matching and residual compensation. Additionally, if compression performance is the goal, the encoder rules will be able to select whether or not to accept the color correction results or to simply encode the original content for a particular block. For instance, in Ref. [48] the DC coefficients at macroblock (MB) level are refined by taking into account the corresponding MB in the reference camera. Another example on DC coefficient modification scheme is introduced by Lee et al.[49], and it has been adopted in the standard MPEG Joint Multiview Video Model (JMVM) reference software. Also in order to improve the interview motion prediction, Yamamoto et al.[50] built a correction lookup table when encoding the video. Although the inside encoder color correction processing can benefit from some intermediary information (e.g., the motion vectors and residual matching) generated by the encoder, poor correction by block-level matching and compensation would result in highly complex adjustment of the whole pipelines in both the encoder and decoder sides.

Postprocessing after decoding. Several works perform color correction as a postprocessing phase for the decoded video streams. For example, inspired from image-based colorization[11], a color annotation strategy is proposed in Ref. [51] for block-based color fusion optimization. Nevertheless, in MB-based color fusion it is still difficult to avoid blocking artifacts or over blurring effects, due to the lack of an accurate fusion criterion.

Color compensation strategies. A critical task in multiview color correction is to compensate for color differences between multiple views of the same scene. Existing color compensation methods in this domain include using a single scaling factor[49], low-dimensional linear matrix or combined linear matrices[36, 38, 43, 44], Pairwise basis function[50], accumulative histogram matching[37, 45], or high-order polynomials[39]. Those methods using a single scaling factor can perfectly map the average color between the target and reference images, but they easily suffer from over-saturation. In order to better handle linear color scaling, linear matrix transformation based solutions are proposed in Refs. [40, 44, 46, 51] for different color components. Such linear transform matrices, usually optimally solved by an over-determined linear system, support flexible scaling operations. On the other hand, they usually operate on 4 × 4 blocks, yielding limited degrees of freedom. Pairwise basis function based methods are suitable to fit various discrete processing units of image segmentation[52] or separated Gaussian model[53], but accurate video segmentation is still difficult. Moreover, even if with good segmentation results, color compensation on segmented areas or blocks would result in outliers and the subsequent color compensation may introduce blocking artifacts or obvious color gaps between different areas. To address this problem, Lu et al.[35] recently proposed to maintain the original local texture information for each pixel, and the *global* color compensation is formulated and solved using a sparse Laplacian matrix based optimization (see the results in Fig. 3). However, robust color compensation for large baseline camera views under complex lighting conditions is still an open issue.

Correspondence construction. Correspondence matching for different views is another key issue in multiview color correction. Features in consecutive frames in the same view as well as in synchronized frames from different views are supposed to appear similar. It is therefore important to be able to match features between these different images. In the literature, the involved matching methods can be classified as sparse or dense matching. For the sparse matching methods, Scale-Invariant Feature Transform (SIFT) is employed in Refs. [41, 44–46, 54, 55], while Speeded-Up Robust Features (SURF) has also been used in Refs. [35, 47]. In Ref. [45], RANSAC is further employed to remove the outliers of the feature points. One of the disadvantages of sparse matching lies in relatively few detected points, such that the color mapping would not be well performed for all image pixels. Thus, dense matching methods have also been introduced so as to overcome this drawback. In this class of methods, researchers attempt to use optical flow[47], disparity estimation[43], block matching[40],

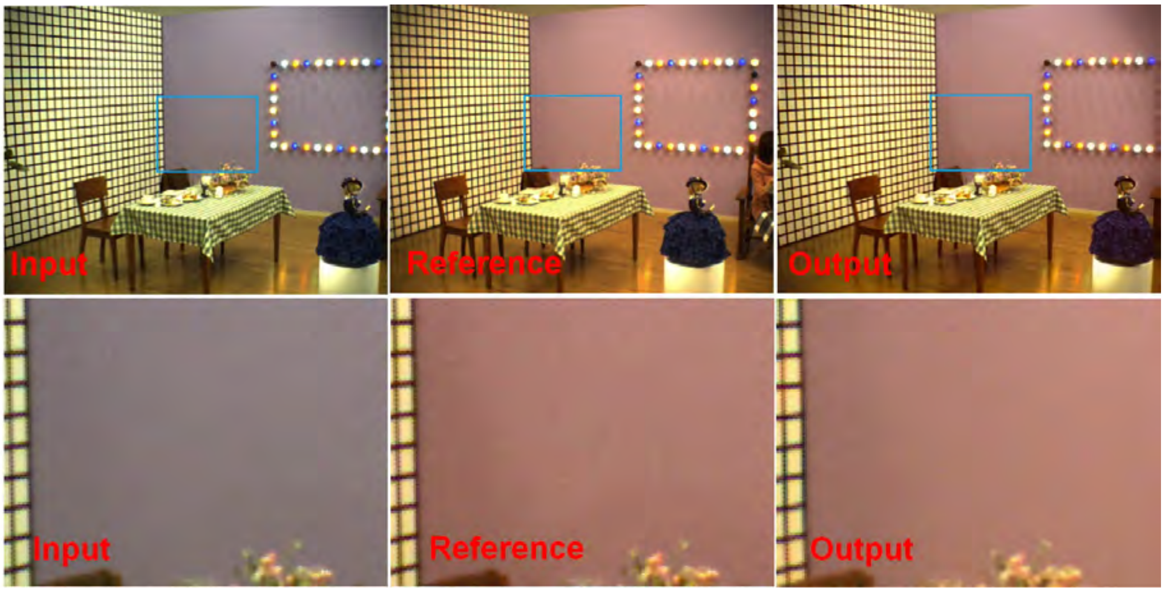

Fig. 3 Color correction for multiview video. Examples are from Ref. [35].

and even pre-segmented local matching[44, 52]. In Ref. [44], the detected sparse feature points are further used to guide the matching by combining them with their corresponding local regions. As mentioned early, robust multiview video segmentation involved in dense matching is still very challenging. Moreover, dense matching based correction methods using blocks or segmented areas would result in obvious overlapping between matched areas (or blocks), making color compensation strategies much more complex.

Evaluation. Most of existing color correction methods take a given video as the reference and the video captured by the central camera of the acquisition system is usually chosen as the reference. This is under the assumption that when the scene appears natural and consistent, other views share the most parts of content with the central view. But in some cases it is unreasonable to follow it if the central view is color distorted or even the cameras are arranged in dome, circular, or other special camera arrangements. Therefore, several other approaches attempt to find better reference for all input cameras. For instance, in Ref. [45], the optimal reference view is selected by evaluating the histogram differences between different views. In Ref. [44], the mean value of a small window in all corresponding views is used to maintain the color consistency. Similar strategies can also be seen in Ref. [39, 46]; the former method directly computes the average color of all views, while the latter takes the mean color, obtained from those identified common corresponding points on the computed temporal SIFT-flow, as the reference color.

Objective or subjective evaluation of the color correction is an interesting but also difficult issue. Concerning objective evaluation, the researchers usually employ the well-known Structural SIMilarity (SSIM) metric to evaluate the reconstructed structure information, and the Peak-Signal-to-Noise-Ratio (PSNR) is employed to calculate the reconstructed color images. For example, Xu and Mulligan[56] used the PSNR to evaluate the overlapped area for the reference image and the color corrected one, and calculate the SSIM as the structure similarity metric between the target image and its color correction version. Nevertheless, in the first step the overlapped area pairs may not be perfectly matching between each other, and the second step lacks the color transfer evaluation. Even worse, although various real or virtual views have been chosen as the reference, because multiview videos are always captured by different cameras, and the real scene exhibits complex lighting and reflection conditions, no ground truth is available yet in this domain, and thus simply calculating the PSNR or SSIM as employed in Ref. [56] is not the most appropriate approach to investigate the color correction effect. To address it, in Ref. [54], a distortion function using the gamma curve and linear transfer was proposed. However, it is difficult to fit the uncorrected colors between different views by just using a linear transfer model. Recently, a forward-reverse evaluation model was presented in Ref. [35]. As shown in Fig. 4, the input video is firstly color corrected to match a particular reference. Then, this result is inversely corrected by taking the original source video as a reference color hereby attempting to undo the initial or forward correction. After that, PSNR and SSIM metrics can show the quality of the twice processed frame with respect to the original, hereby quantifying any structural distortions that could have been introduced.

Regarding subjective evaluation of color correction results, we note that this area is still far from being mature. Subjective viewing tests, e.g., by applying the color discrepancy model[57] for stereoscopic content, would be a potential solution.

3 Multiview Video Synthesis

3.1 Virtual view synthesis

In various multiview-oriented applications, it is common to re-render an image as if it would have been captured by another camera. Obvious examples are advanced immersive 3-D systems that enable free navigation or that want to visualize dense multiview 3-D content from a sparse set of input views[58, 59]. In these examples, the synthesized imagery is presented to an end user, however, sometimes view synthesis can be of use as a building block in some other computational

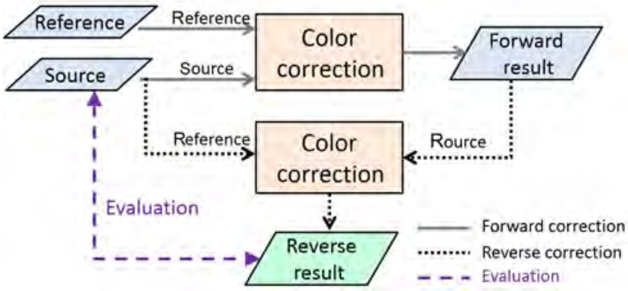

Fig. 4 Forward-reverse evaluation[35] for multiview color correction.

pipeline. In the coding of multiview video, for example, view synthesis has already been introduced to improve prediction strategies[60, 61] between different views. This improved predictions enable the encoder to compress the multiview content more efficiently compared to an encoder that does not exploit the inter-view correlation. Additionally, rectification and removal of lens distortions can also be considered as forms of view synthesis, even though the viewpoint may not change as dramatically compared to the other scenarios.

3.1.1 Warping

When the content is available in the form of a textured mesh, any viewpoint can be rendered using a computer graphics renderer like the ones used in computer games. However, if the content is represented as a set of images from a (sparse) set of viewpoints, the problem is much more challenging. Algorithms that tackle this are referred to as Image-Based Rendering (IBR) algorithms. When in addition to color images, depth maps are also available, there are the so-called DIBR methods. We focus on the latter.

In order to synthesize a novel viewpoint, we assume that a depth map is available and the camera is calibrated. Originally, a point $[x, y, z, 1]^{\mathrm{T}}$ (in homogeneous coordinates) is warped to the pixel location $[u, v, 1]^{\mathrm{T}}$ by the projection equations:

$$[u', v', z_{\mathrm{c}}]^{\mathrm{T}} = \boldsymbol{K} \cdot [R|t] \cdot [x, y, z, 1]^{\mathrm{T}} \tag{1}$$

$$[u, v, 1]^{\mathrm{T}} = [u', v', z_{\mathrm{c}}]^{\mathrm{T}} / z_{\mathrm{c}} \tag{2}$$

where $[R|t]$ is the 3×4 extrinsic matrix which transforms a point from world coordinates to camera coordinates, and $\boldsymbol{K}$ is the camera's 3×3 intrinsic matrix which expressed how a point in camera coordinates is transformed to pixel coordinates. z_{c} is the z-coordinate of the point with respect to the camera's axes. The final projection is performed by dividing by z_{c} and we obtain the image point $[u, v, 1]^{\mathrm{T}}$. The projection is generally only invertible up to a scaling factor, but this scaling factor is exactly given by the depth map.

When re-projecting the multiview image pixels to 3-D world coordinates, they essentially form a pointcloud which can then be rendered from another viewpoint by projecting them on the pixel grid of the desired camera and maintaining a z-buffer to determine which pixels should be visible. A naive implementation would thus apply so-called forward warping and directly warp both the color and depth information from the source image to the target viewpoint. However, due to discretization of the pixel coordinates and inaccuracies in the depth maps, this may lead to missing pixels and rendering artifacts (This can be seen in Fig. 5.). One potential solution to splat projected points over multiple pixels instead of simply rounding off to a single pixel. As this can cause over-blurring, a better approach is to firstly warp the depth map from reference to source (forward warp) and then use this depth map in order to sample the colors from the reference image (backward warp). The advantage of this two-step warping is that the virtual depth map can first be cleaned by applying various filtering or denoising algorithms before sampling the colors. This can significantly reduce the number of artifacts in the synthesized image.

3.1.2 Disocclusion handling

When changing the viewpoint on a scene, areas that were previously occluded by some foreground object may become visible. This is referred to as disocclusion and results in empty areas surrounding objects in the

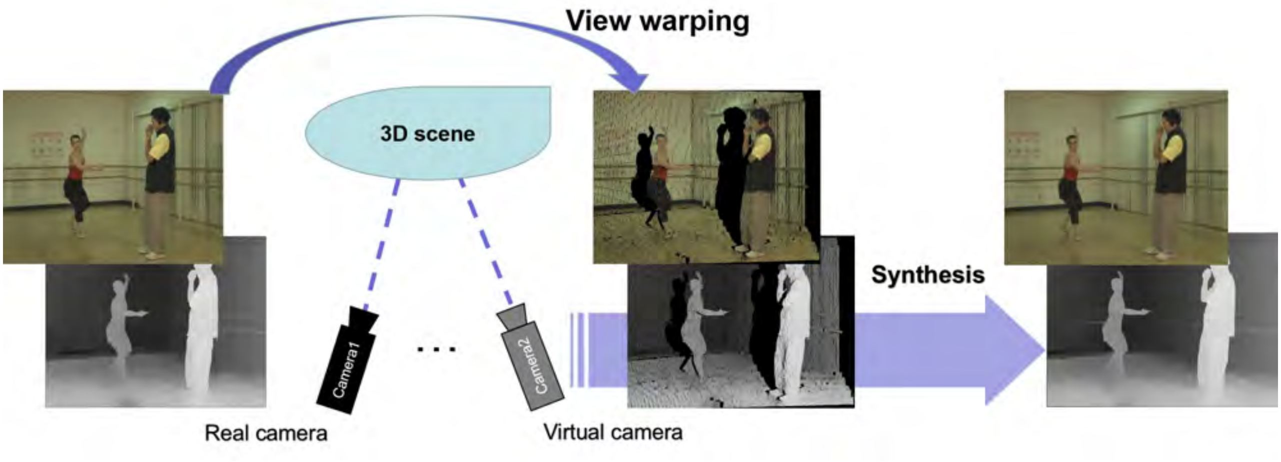

Fig. 5 Warping-based novel viewpoint synthesis using multiview video plus depth maps.

synthesized image (see Fig. 5). There are generally two ways to fill in disoccluded pixels. Some researchers make sure the virtual depth map contains no holes. This means that for every virtual pixel, a color can be found in one of the camera. Others have developed specialized inpainting algorithms, usually inspired by those in single image/video inpainting[62] but taking into account depth information.

Avoiding holes. In Ref. [63], a regular grid is imposed on the depth map. The warping procedure is then implemented as a deformation of this grid in such a way that no holes are created. This means that in the backward warp of the two-step process, a color can be found for every pixel in the virtual image. A similar idea in order to prevent holes is by firstly trying to estimate the full depth map of the desired virtual view and then use this depth map as a guide to sample colors from the reference views. In Ref. [64], this is implemented by so called plane-sweeping. Every pixel in the virtual view is assigned a tentative depth value. Based on this value, the pixel can be warped on multiple reference images and a cost is computed based on how much the references agree or disagree about the color for a particular pixel. Next, the depth value is increased and a new cost is computed. In the end, each pixel in the virtual view is assigned the depth and color on which most of the references agreed. This method is inherently massively parallel and therefore very suited for real-time GPU implementation. The plane sweeping method has been shown to deliver good results for free navigation in soccer video[65]. However, when the scene contains complicated textures or the displacement of the virtual camera is too large or non-linear methods like Refs. [63, 64] will result in blurry areas and ghost edges.

Inpainting. A more common approach for disocclusion handling is to employ inpainting algorithms. There exist a lot of methods to erase content from a still 2-D image or even 2-D video. In Ref. [66], object tracking and reconstructed homography are further introduced to preserve temporal coherence. However, directly applying these methods to erase disocclusion areas will generally not generate satisfactory results. Remember that most patch-based inpainting methods follow a kind of onion peeling approach where iteratively, a region of the hole boundary is filled in by a patch that resembles the overlap with the already known area. The filling order is driven by the presence of structures in the image. Disocclusion regions however have a clear physical origin and it is known that they should be filled in with only patches that are sampled from the scene's background. Therefore, various view synthesis algorithms in the literature adapt a classical 2-D image inpainting algorithm in order to make them depth aware and avoid the bleeding artifacts that would otherwise occur (see Fig. 6).

The first class of existing inpainting methods is based on interpolation or diffusion. Smaller holes may be filled in using either Gaussian filtering or median filtering, while for larger holes an iterated diffusion process makes sure that strong contours are extended in the disocclusion hole[67]. These kinds of methods are simple and efficient for smaller disocclusion and content with simple textures.

For larger disocclusion holes, most researchers tend to go for a patch-based method, usually based on Criminisi et al.'s work[68]. In Ref. [68], the border of the hole is referred to as the fill front $\partial\Omega$ (see Fig. 7). In every iteration, the pixel p of the highest priority $P(p)$ is selected and a patch $\psi(p)$ around it is extracted. Since $p \in \partial\Omega$, $\psi(p)$ overlaps with the known region of the image. Based on this overlap, a patch $\hat{\psi}(p)$ is found such that the sum of squared differences $\mathrm{SSD}(\psi(p), \hat{\psi}(p))$, computed only on the known pixels of $\psi(p)$, is minimized. The priority function $P(p)$ is constructed in such a way that patches that extend edges are favored over others. Because of their use of patches and the clever choice of $P(p)$, the method of Ref. [68] is able to preserve both the texture and structure of the image.

Daribo and Pesquet-Popescu[69] extended Ref. [68]'s method to inpainting disocclusions by refining calculation of $P(p)$ and the search for $\hat{\psi}(p)$ by taking the depth information into account. Similarly, Gautier et al.[70] used a tensor-based structure propagation approach to refine the priority of structural textures

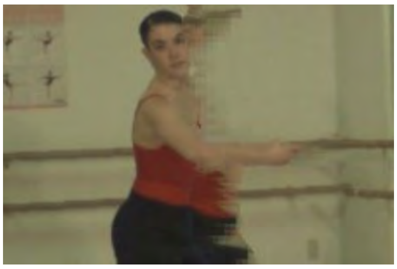

Fig. 6 Bleeding of foreground object when classic 2-D image inpainting is used to fill the disoccluded area after 3-D warping.

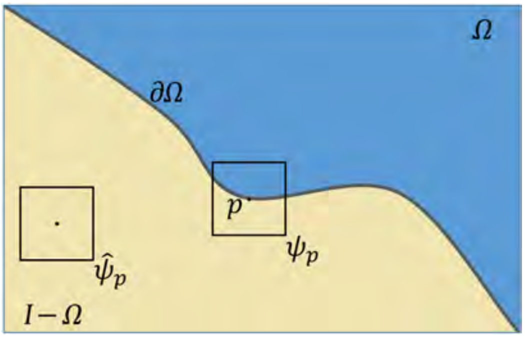

Fig. 7 Inpainting schema: the region *Ω* needs to be filled by copying and pasting patches from the known region *I*–*Ω*.

based on their local geometry-based inpainting strategy[55]. In Ref. [71] a Hessian matrix structure tensor is presented to construct a more robust match priority of Ref. [68]. In these methods, depth-based foreground and background analysis can be used to guide the inpainting processing with reasonable constraints[72, 73] (see an example in Fig. 8).

The main disadvantage of Ref. [68] and its extensions is that it is a greedy method and cannot backtrack if it at one point makes a wrong decision. In Ref. [74], the inpainting problem was posed as an energy minimization problem on a 2-D Markov Random Field (MRF). Reference [74] proposes a new variant of the classical Belief Propagation (BP) algorithm and names it priority-BP. However, the original 2-D inpainting method of Ref. [74] is still relatively slow even if with clever implementations that use frequency domain computations and multi-scale processing. Moreover, like any regular 2-D inpainting method, it generates artifacts by bleeding pixels from foreground objects into the background when applied to fill disocclusioins. In Ref. [75], the method is essentially extended by disabling the edge of the MRF that lies on the boundary of a foreground object and by incorporating depth information in the cost function. While avoiding the bleeding artifact, the method is not stated to have gained a significant speedup. In Ref. [76], an additional extension is proposed that limits the number of patches that need to be evaluated per node in the MRF, which greatly reduces computation time. The method of Ref. [77] further builds these observations. This proposed inpainting method pays more attention to the camera movement and improves the optimal candidate selection for the final optimization by also considering the depth information around the disoccluded areas. By constructing a simple but intuitive priority-function that promotes the propagation of background pixels, the priority-BP algorithm is accelerated.

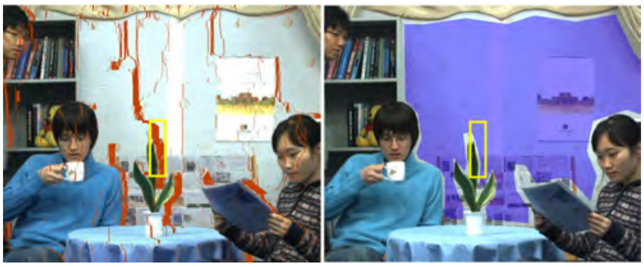

Fig. 8 Depth aware candidate search for patch-based inpainting, with which reference patches of the disoccluded areas (see the yellow rectangle) are preferred from the background (see the purple area in the right sub-figure).

Computational complexity. Most of the computation time is spent on comparisons of image patches. Patch-based methods usually have to look for a fully known patch that would fit the already known pixels around the disocclusion border. Exhaustive search in high resolution video is too slow. Some methods such as Refs. [69, 70, 77] use the depth map in order to limit the region that needs to be searched. Other works apply fast approximate nearest neighbor algorithms such as PatchMatch[78], kd-trees[79] or PatchTable[80]. Because of the efficiency on approximate nearest-neighbor match, PatchMatch has been applied in stereo matching[81], semantic segmentation[82], and content completion of stereo image pairs[83]. Moreover, a pixel-level multiview video inpainting method[84, 85] is introduced based on it (see Fig. 9). This technique worked multiscale and calculates nearest neighbor fields to find the optimal matching candidates for each pixel.

Temporal consistency. If inpainting is performed on the same region in successive frames, it often creates a flickering effect when playing the sequence as a video. This can be due to the randomness in some algorithms or due to noise in the depth maps. Ndjiki-Nya et al.[86] proposed to build a sprite model of the scene's background. As foreground objects in the scene move, occluded parts become visible and are added to the sprite. Now, when a novel viewpoint needs to be synthesized, the algorithm first checks whether the sprite contains the required information and only resorts inpainting when it does not. To compensate for illumination changes that may occur over time, Ref. [86] employs the seamless cloning method of Ref. [87]. In Ref. [88], holes are classified as either static or dynamic using optical flow. By only inpainting

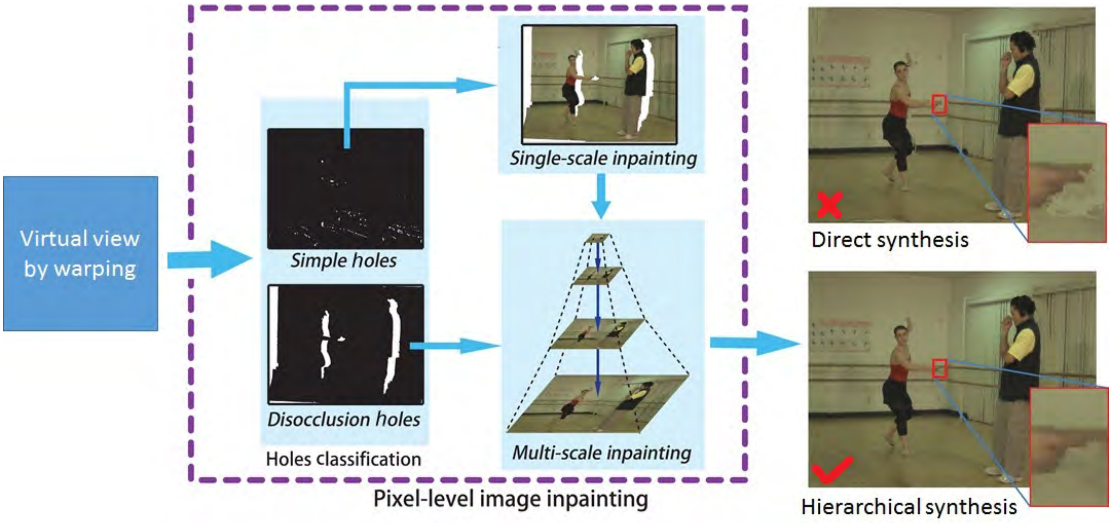

Fig. 9 Multi-scale texture inpainting for multiview video syntheisis[84].

static holes once, temporal consistency is maintained. In Ref. [89], a background modeling based on Gaussian Mixture Models (GMM) is proposed. They further include motion estimation and homography estimation based on sparse features to compensate for camera movement.

3.1.3 Open problems

A lot of research has been done on view synthesis. However, most of the resulting publications focus on linear transitions between two cameras that are relatively close together. To maintain temporal consistency, it is often assumed that the scene background is static. Future research will likely address the difficulties of handling with (1) large baselines, (2) non-linear camera paths, (3) non-horizontal camera motion, and (4) background motion.

Furthermore, latest research focuses include deep learning-based view synthesis, for example, combining MRF and Convolutional Neural Networks[90], and 3-D model assisted view synthesis[91–93]. Indeed, when combining with image datasets or even Internet images (see the faithful completion[94] in Fig. 10), such advanced approaches greatly enrich existing research directions on view synthesis of multiview video.

4 Multiview Editing and Interaction

The availability of multiple view of the same scene makes it possible to recover the underlying 3-D geometric structure of the scene. This information will promote the use of object-level editing in multiview video systems. In this section, we focus on methods exploiting geometric information about the underlying scene when two or more views are available. We also review some object-level editing on single image/video that could be generalized to multiview videos in the future.

4.1 Single image/video editing

From the view of geometric structure understanding, traditional editing methods can be grouped into two categories, i.e., pixel-level editing and object-level editing. Pixel-level editing performs pixel-wise manipulations (color, position, etc.) as Photoshop does, while object-level editing focuses on more semantic operations (e.g., roll over the red car in the image) with geometric analysis of the scene. In this section, we focus on the progress in object-level editing for single 2-D images and videos.

Objects in a single image are usually modeled and manipulated using 3-D model proxies. Common object-level editing operations are then supported, such as 3-D scaling, rotation, and translation of a particular object in an image/video. This is contrast to classical operations such as filtering, recoloring, inpainting and copy-and-paste.

Hornung et al.[95] animated 2-D characters in a still 2-D image. They proposed to match 3-D motion-captured data of a human actor's skeleton. They fit a skeleton on the 2-D character and searched for initial pose of the character in the motion sequence. After that, an animation is generated using shape deformation in the image space guided by the projected joint positions of the skeleton in the motion sequence. Zheng et

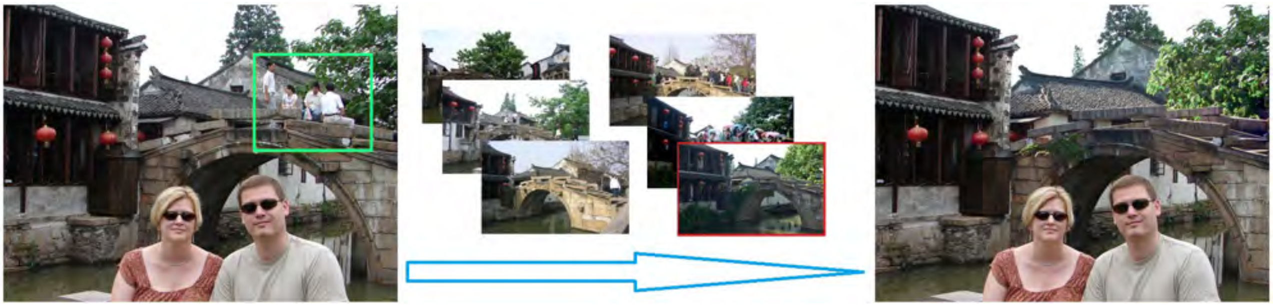

Fig. 10 Faithful image completion using Internet images. Examples are from Ref. [94].

al.[96] manipulated scene objects using cuboid proxies, i.e., operating an object means operating their cuboid proxy. With the objects of interest annotated by a user, they divide the input image into a background layer and textured 3-D cuboid proxies. The scene's geometry (occlusion, orientation, coplanarity, etc.) is also analyzed, which in turn enables some smart object-level manipulations, e.g., translation, rotate, and deformation. Unlike the simple cube model used in Ref. [95], Chen et al.[97] proposed to represent the 3-D model of the objects of interest by a generalized cylinder, cuboid or similar primitives, which covers a larger range of man-made objects. The model is extracted using just three strokes, called *3-sweep*, which generates a 2-D cross profile of the model using the first two strokes and sweeps along a straight or curved axis to fit the final 3-D model using the third stroke. With a stock 3-D model database available, Kholgade et al.[98] aligned a best 3-D model to the object in the 2-D image. They completed the hidden parts of the image objects by leveraging symmetries and the stock model appearance, which supports more powerful 3-D operations, e.g., flip, on objects of interest.

Instead of recovering the exact geometries of objects, Hoiem et al.[99] learned a statistical model of geometric labels, which decompose the outdoor scene image into three parts, i.e., "ground", "sky", and "vertical". Given an input outdoor scene, the horizontal positions of objects, i.e., "vertical", can be estimated from the geometric labels of the image predicted by the statistical model.

Although some pixel-level editing operations can be directly applied to videos, e.g., color edit propagation[17], applying object-level editing to video is usually more elaborate with spatiotemporal constraints. Lu et al.[100] proposed to edit individual objects along their time-lines in videos (see Fig. 11). An individual object is kept at its original spatial location, however, it may appear at a different time, thus resulting in new temporal relationships between objects, which enables object-level effects, such as slow motion, fast motion, and even time reversal effects, as shown in Fig. 11. Zhang et al.[101] proposed a method to cutout videos, i.e., extracting moving objects from videos, with less user interaction and fast feedback by exploring spatiotemporal relationship of neighboring patches from video frames and further optimizing the object extraction using graph cut. In order to remove the distraction objects during video stabilization, Zhang et al.[102] proposed to plan the new camera path to avoid distraction objects. A two-pass optimization is exploited to achieve this goal. In the first pass, an initial smooth camera path which avoids distractions and keeps as much significant original content as possible is obtained. Then the initial camera path is further segmented to shorter paths with simple models, e.g., zoom, rotation, and translation. These models are linearly fitted, followed by a second optimization aiming to eliminate undesired contents while smooth the final path to the fitted models across all segments.

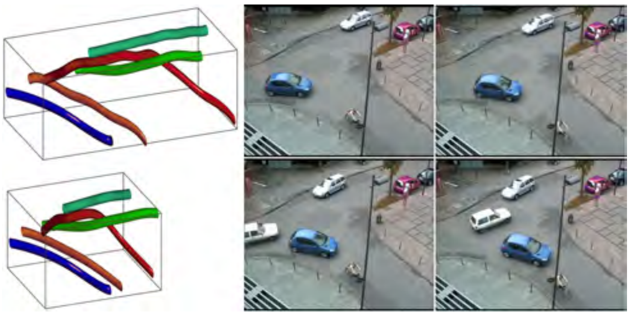

Fig. 11 Time-line editing for video objects. Spatio-temporal trajectories of objects are extracted and rearranged to produce outputs with different time effects. Examples are from Ref. [100].

4.2 Multiview video editing

A special kind of multiview, so called *stereoscopic 3-D* *vision* or just *stereo*, contains only two slightly different views, which produces horizontal displacements, called *retinal disparities*, between corresponding points in the retinal images of eyes. The corresponding difference in a 2-D image pairs is called *pixel disparity* (referred to as disparity afterwards) accordingly, resulting in depth perception when stereoscopically viewed.

Compared to the single image/video, disparities from stereo correspondences provide information about the depth distribution of the scene. So, processing the stereoscopic data requires careful attention on disparities, which would result in wrong depth perception if improperly manipulated. More importantly, improper stereoscopic content may cause visual fatigue and discomfort[103], reducing the visual experience of 3-D. Some perception models on disparity[104, 105] and motion of disparity[106–108] have been proposed to guide the manipulation of stereoscopic content to ensure the disparity lying in the zone of comfort[109]. Thus, how to remap the original disparities to a new range of disparity while keeping the right stereopsis in stereoscopic images becomes the fundamental part in all stereoscopic applications, such as retargeting[110–114], warping[115–119], completion[120], etc.

Lang et al.[110] adjusted the disparities to a new rang according to user specified nonlinear maps. In order to display original stereoscopic content on other devices, seam carving[121] on single 2-D image has been extended to stereo image pairs[111]. Lee et al.[112] proposed to resize stereoscopic images according to different layers of depth and colors. Chang et al.[113] kept salient objects in the comfort zone of displays while retargeting stereo content.

A typical processing pipeline of stereoscopic image warping from Ref. [118] is presented in Fig. 12. The authors aim to paste a 2-D source into a target stereoscopic scene. Firstly, the underlying disparities of source and target are estimated. Then the source disparities are remapped to be consistent with their surroundings in the source scene. The final composition results are generated using traditional 2-D image warping methods with the stereo correspondences constraints guided by the underlying disparities. On the other hand, Lo et al.[115] and Luo et al.[116] both selected stereo content of interest somewhere else and composited it into new positions with seamless warping. Niu et al.[117] adapted 2-D image warping to stereo images. Du et al.[119] generated images under new viewpoint using feature correspondences and line constraints between the stereo image pair. Disparity information is also considered in optimizing stereo patch-based synthesis[114, 120] to enable a wide variety of stereoscopic image editing applications, such as texture synthesis, content adaptation, and inpainting.

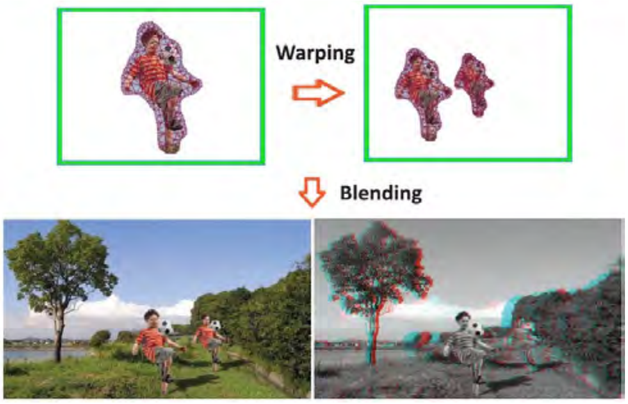

Fig. 12 Warping-based stereoscopic image editing. Examples are from Ref. [118].

Some editing tools on 2-D images/videos have also been extended to stereoscopic videos with additional constraints on stereo consistency. Taking the spatiotemporal stereo consistency into account, Raimbault and Kokaram[122] reduced the amount of candidate pixels to search for stereo video inpainting when using exemplar-based techniques. Liu et al.[123] found out that the low-rank subspace constraint for monocular vidoes[124] holds for stereoscopic videos and stereoscopic video stabilization can be achieved without explicit 3-D reconstruction. Kopf et al.[125] retargeted stereoscopic videos considering content saliency and map warps from left view to right view to guarantee the stereo consistency. The perceptual comfort zone of disparity is also considered in Ref. [126]. Recently, Wang et al.[127] provided a framework to automatically adjust the depth of stereoscopic video to the zone of comfort using perceptual models on disparity, motion of disparity, stereo window violation, etc.

Compared to the only two views in stereoscopic 3-D vision, more views of the same scene will help to reconstruct a more accurate 3-D geometry of the scene. Jiang et al.[128] estimated a 3-D pointcloud from multiple views by *structure-from-motion*. These 3-D points together with color and texture information of each view are later used to identify multiple groups of repetitive structures, e.g., windows and balconies. Their method can detect repetitive structures on curved surfaces and in turn consolidate the underlying point cloud. With the help of reconstructed sparse 3-D point cloud of the scene, Laffont et al.[129] obtained a rich intrinsic image decomposition for outdoor scenes with reflectance, sum illumination, sky illumination, and indirect illumination. Djelouah et al.[130] applied graph cut to multi-view images/videos to segment out objects of interest. Segmentation information is propagated between viewpoints via the projections of a sparse 3-D sampling to each view and temporally evolved along both an optic flow and a SIFT flow.

Graph models have also been exploited in some other multiview video editing. Fu et al.[131] presented a method to summarize multiview videos. Original multiple videos are divided into different shots, on which a spatiotemporal shot graph considering temporal consistency and content similarity is built. Then, the summarization is generated by clustering similar shots favoring interesting event via random walks and solving a multi-objective optimization. Their method results in different objective summarization, such as minimum length, maximum informative coverage, and multi-level summarization. Shamir et al.[132] focused on the gaze center of attention, called *3-D joint attention*, of social cameras which are capturing a same activity and provide a single coherent “cut” video of the activity. A trellis graph takes the joint attention in frames of each camera as nodes and connects edges between the nodes, representing transition of cameras. A path from start time to the end time in the graph defines an output movie and the best “cut” considering the cinematographic guidelines is optimized using dynamic programming. Wang et al.[133] proposed to temporally align two videos taken from the same scene at different times by computing a minimum path in a graph defined by content similarity of all pairs of frames from the two videos. The globally optimal temporal alignment of multiple videos is achieved via the minimum spanning tree of the graph induced by the pairwise alignment cost.

4.3 Open problems

Current multiview video editing methods are mainly focused on the analysis of relationships among contents from different views. These methods[130–133] usually use a graph to model the relationships. However, explicit 3-D geometries of objects are not reconstructed in these scenarios, thus, these methods are less capable of supporting object-level editing as in 2-D images[96–98].

Although a precise 3-D reconstruction from single image/view is still a challenge in computer vision, multiview videos provide much rich spatiotemporal structure information, which would facilitate geometric reconstruction. With depth images available, it would be more convenient to understand the geometric structure of scene[134].

5 Discussion and Conclusion

DIBR-based multiview video synthesis and editing aims to effectively support various potential solutions and applications on the full chain of 3-D free viewpoint TV displays. In this paper we have discussed the state-of-the-art DIBR-based multiview video processing, and particularly we focused on several key topics on multiview video-based synthesis, and interaction, i.e., color correction, virtual view synthesis, and interactive editing. Besides that, corresponding techniques of these topics on single image/video-based processing are reviewed.

The ongoing advances on image/video-based processing, such as deep learning and 3-D dataset analysis and modeling, are rapidly boosting almost all of the relevant issues on multiview video processing. As has been shown in the latest literature, when such modern and powerful techniques combined with multiview video resources, great potential could be further exploited. The foreseeable future work should consider synthesis and editing using semantic understanding (e.g., Refs. [134, 135]), shape extraction (e.g., the 3-Sweep interaction[97] in Fig. 13) and

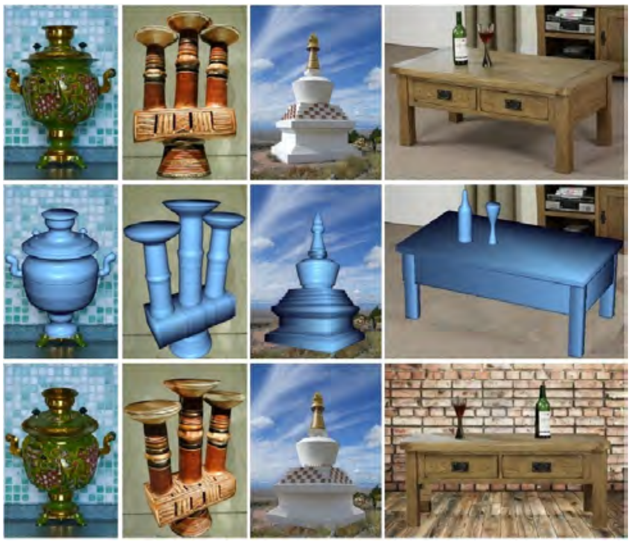

Fig. 13 3-Sweep: Object-level editing in 2-D images is enabled after 3-D models are extracted with user interactions. Examples are from Ref. [97].

efficient learning-based training models. Besides that, the incorporation of view synthesis along with highly intelligent compression and transmission strategies (e.g., 2-D texture preservation[136] and 3-D distortion minimization[137]) is one of the potential avenues for future work. We believe that with the progress of such novel techniques on multiview video processing, more and more free viewpoint TV displays and 3-D immersive interaction-oriented applications can reach the consumer and prosumer markets in the near future.

Acknowledgment

Shaoping Lu was partially supported by Innoviris (3-DLicornea project) and FWO (project G.0256.15). This work was also supported by the National Natural Science Foundation of China (Nos. 61272226 and 61373069), Research Grant of Beijing Higher Institution Engineering Research Center, Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, and Tsinghua University Initiative Scientific Research Program.

References

[1] A. A. Alatan, Y. Yemez, U. Gudukbay, X. Zabulis, K. Muller, I. E. Erdem, C. Weigel, and A. Smolic, Scene representation technologies for 3DTV—A survey, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 17, pp. 1587–1605, 2007.

[2] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, Multiview imaging and 3DTV, *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 10–21, 2007.

[3] C. Fehn, Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV, in *Electronic Imaging*, 2004, pp. 93–104.

[4] W. Sun, L. Xu, O. C. Au, S. H. Chui, and C. W. Kwok, An overview of free view-point depth-image-based rendering (DIBR), in *Proc. APSIPA*, 2010, pp. 1023–1030.

[5] A. Mustafa, H. Kim, J.-Y. Guillemaut, and A. Hilton, General dynamic scene reconstruction from multiple view video, in *Proc. ICCV*, 2015, pp. 900–908.

[6] Middlebury multiview stereo, http://vision.middlebury.edu/mview/, 2016.

[7] A. Vetro, T. Wiegand, and G. J. Sullivan, Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard, *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[8] D. Ren, S.-H. G. Chan, G. Cheung, V. Zhao, and P. Frossard, Anchor view allocation for collaborative free viewpoint video streaming, *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 307–322, 2015.

[9] A. Gijsenij, T. Gevers, and J. Van De Weijer, Computational color constancy: Survey and experiments, *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2475–2489, 2011.

[10] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, Color transfer between images, *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, 2001.

[11] A. Levin, D. Lischinski, and Y. Weiss, Colorization using optimization, *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[12] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, Interactive local adjustment of tonal values, *ACM Trans. Graph.*, vol. 25, pp. 646–653, 2006.

[13] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, Edge-preserving decompositions for multi-scale tone and detail manipulation, *ACM Trans. Graph.*, vol. 27, p. 67, 2008.

[14] X. Xiao and L. Ma, Gradient-preserving color transfer, *Comput. Graph. Forum*, vol. 28, no. 7, pp. 1879–1886, 2009.

[15] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, Gradientshop: A gradient-domain optimization framework for image and video filtering, *ACM Trans. Graph.*, vol. 29, no. 2, p. 10, 2010.

[16] X. An and F. Pellacini, Appprop: All-pairs appearance-space edit propagation, *ACM Trans. Graph.*, vol. 27, no. 3, p. 40, 2008.

[17] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, Efficient affinity-based edit propagation using kd tree, *ACM Trans. Graph.*, vol. 28, no. 5, p. 118, 2009.

[18] Y. Li, T. Ju, and S. Hu, Instant propagation of sparse edits on images and videos, *Comput. Graph. Forum*, vol. 29, pp. 2049–2054, 2010.

[19] X. Chen, D. Zou, Q. Zhao, and P. Tan, Manifold preserving edit propagation, *ACM Trans. Graph.*, vol. 31, no. 6, p. 132, 2012.

[20] S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[21] L.-Q. Ma and K. Xu, Efficient manifold preserving edit propagation with adaptive neighborhood size, *Computers & Graphics*, vol. 38, pp. 167–173, 2014.

[22] L. Xu, Q. Yan, and J. Jia, A sparse control model for image and video editing, *ACM Trans. Graph.*, vol. 32, no. 6, p. 197, 2013.

[23] Z. Cheng, Q. Yang, and B. Sheng, Deep colorization, in *Proc. ICCV*, 2015, pp. 415–423.

[24] S. Iizuka, E. Simo-Serra, and H. Ishikawa, Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification, *ACM Trans. Graph.*, vol. 35, no. 4, p. 110, 2016.

[25] M. Afifi and K. F. Hussain, Mpb: A modified poisson blending technique, *Comp. Visual Media*, vol. 1, no. 4, pp. 331–341, 2015.

[26] Y. Qian, D. Liao, and J. Zhou, Manifold alignment based color transfer for multiview image stitching, in *Proc. ICIP*, 2013, pp. 1341–1345.

[27] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, Optimizing color consistency in photo collections, *ACM Trans. Graph.*, vol. 32, no. 4, p. 85, 2013.

[28] J. Park, Y.-W. Tai, S. N. Sinha, and I. S. Kweon, Efficient and robust color consistency for community photo collections, in *Proc. CVPR*, 2016, pp. 430–438.

[29] Z. Farbman and D. Lischinski, Tonal stabilization of video, *ACM Trans. Graph.*, vol. 30, no. 4, p. 89, 2011.

[30] Y. Wang, D. Tao, X. Li, M. Song, J. Bu, and P. Tan, Video tonal stabilization via color states smoothing, *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4838–4849, 2014.

[31] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, Blind video temporal consistency, *ACM Trans. Graph.*, vol. 34, no. 6, p. 196, 2015.

[32] Y. Lei, W. Luo, Y. Wang, and J. Huang, Video sequence matching based on the invariance of color correlation, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1332–1343, 2012.

[33] S.-P. Lu, G. Dauphin, G. Lafruit, and A. Munteanu, Color retargeting: Interactive time-varying color image composition from time-lapse sequences, *Comp. Visual Media*, vol. 1, no. 4, pp. 321–330, 2015.

[34] J. Zhong, B. Kleijn, and X. Hu, Camera control in multi-camera systems for video quality enhancement, *IEEE Sensors Journal*, vol. 14, no. 9, pp. 2955–2966, 2013.

[35] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, Spatio-temporally consistent color and structure optimization for multiview video color correction, *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 577–590, 2015.

[36] A. Ilie and G. Welch, Ensuring color consistency across multiple cameras, in *Proc. ICCV*, vol. 2, 2005, pp. 1268–1275.

[37] U. Fecker, M. Barkowsky, and A. Kaup, Histogram-based prefiltering for luminance and chrominance compensation of multiview video, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1258–1267, 2008.

[38] Y. Chen, C. Cai, and J. Liu, Yuv correction for multi-view video compression, in *Proc. ICPR*, vol. 3, 2006, pp. 734–737.

[39] C. Doutre and P. Nasiopoulos, Color correction preprocessing for multiview video coding, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1400–1406, 2009.

[40] B. Shi, Y. Li, L. Liu, and C. Xu, Block-based color correction algorithm for multi-view video coding, in *Proc. ICME*, 2009, pp. 65–68.

[41] M. Panahpour Tehrani, A. Ishikawa, S. Sakazawa, and A. Koike, Iterative colour correction of multicamera systems using corresponding feature points, *J. Visual Commun. Image Represent.*, vol. 21, no. 5, pp. 377–391, 2010.

[42] C. Mouffranc and V. Nozick, Colorimetric correction for stereoscopic camera arrays, in *Proc. ACCV*, 2012, pp. 206–217.

[43] F. Shao, G.-Y. Jiang, M. Yu, and Y.-S. Ho, Fast color correction for multi-view video by modeling spatio-temporal variation, *J. Visual Commun. Image Represent.*, vol. 21, no. 5, pp. 392–403, 2010.

[44] K. Li, Q. Dai, and W. Xu, Collaborative color calibration for multi-camera systems, *Signal Process. Image Commun.*, vol. 26, no. 1, pp. 48–60, 2011.

[45] S. A. Fezza, M.-C. Larabi, and K. M. Faraoun, Feature-based color correction of multi-view video for coding and rendering enhancement, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1486–1498, 2014.

[46] H. Zeng, K.-K. Ma, C. Wang, and C. Cai, Sift-flow-based color correction for multi-view video, *Signal Process. Image Commun.*, vol. 36, pp. 53–62, 2015.

[47] B. Ceulemans, S.-P. Lu, P. Schelkens, and A. Munteanu, Globally optimized multiview video color correction using dense spatio-temporal matching, in *Proc. 3DTV*, 2015, pp. 1–4.

[48] H. Jae-Ho, C. Sukhee, and L. Yung-Lyul, Adaptive local illumination change compensation method for h.264avc based multiview video coding, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1496–1505, 2007.

[49] Y.-L. Lee, J. Hur, Y. Lee, K. Han, S. Cho, N. Hur, J. Kim, J. Kim, P. Lai, and A. Ortega, Ce11: Illumination compensation, in *Doc. MPEG & VCEG JVT-U052R2*, 2006, pp. 20–27.

[50] K. Yamamoto, M. Kitahara, H. Kimata, T. Yendo, T. Fujii, M. Tanimoto, S. Shimizu, K. Kamikura, and Y. Yashima, Multiview video coding using view interpolation and color correction, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1436–1449, 2007.

[51] B. Shi, Y. Li, L. Liu, and C. Xu, Color correction and compression for multi-view video using h.264 features, in *Proc. ACCV*, 2009, pp. 43–52.

[52] Q. Wang, P. Yan, Y. Yuan, and X. Li, Robust color correction in stereo vision, in *Proc. ICIP*, 2011, pp. 965–968.

[53] M. Oliveira, A. D. Sappa, and V. Santos, Color correction using 3D Gaussian mixture models, in *Proc. ICAR*, vol. 7324, 2012, pp. 97–106.

[54] K. Yamamoto and R. Oi, Color correction for multi-view video using energy minimization of view networks, *Int. J. Autom. and Comput.*, vol. 5, no. 3, pp. 234–245, 2008.

[55] O. Le Meur, J. Gautier, and C. Guillemot, Examplar-based inpainting based on local geometry, in *Proc. ICIP*, 2011, pp. 3401–3404.

[56] W. Xu and J. Mulligan, Performance evaluation of color correction approaches for automatic multi-view image and video stitching, in *Proc. CVPR*, 2010, pp. 263–270.

[57] M. Plnen, J. Hakala, R. Bilcu, T. Jrvenp, J. Hkkinen, and M. Salmimaa, Color asymmetry in 3D imaging: Influence on the viewing experience, *3D Research*, vol. 3, no. 3, pp. 1–10, 2012.

[58] B. Lee, Three-dimensional displays, past and present, *Physics Today*, vol. 66, no. 4, pp. 36–41, 2013.

[59] H. Urey, K. V. Chellappan, E. Erden, and P. Surman, State of the art in stereoscopic and autostereoscopic displays, *Proceedings of the IEEE*, vol. 99, no. 4, pp. 540–555, 2011.

[60] M. Domanski, O. Stankiewicz, K. Wegner, M. Kurc, J. Konieczny, J. Siast, J. Stankowski, R. Ratajczak, and T. Grajek, High efficiency 3D video coding using new tools based on view synthesis, *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3517–3527, 2013.

[61] F. Zou, D. Tian, A. Vetro, H. Sun, O. C. Au, and S. Shimizu, View synthesis prediction in the 3-D video

coding extensions of avc and hevc, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1696–1708, 2014.

[62] Z. Tauber, Z. Li, and M. Drew, Review and preview: Disocclusion by inpainting for image-based rendering, *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 4, pp. 527–540, 2007.

[63] N. Plath, S. Knorr, L. Goldmann, and T. Sikora, Adaptive image warping for hole prevention in 3D view synthesis, *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3420–3432, 2013.

[64] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, Real-time plane-sweeping stereo with multiple sweeping directions, in *Proc. CVPR*, 2007, pp. 1–8.

[65] P. Goorts, C. Ancuti, M. Dumont, and P. Bekaert, Real-time video-based view interpolation of soccer events using depth-selective plane sweeping, in *Proc. VISAPP*, 2013.

[66] J. Herling and W. Broll, High-quality real-time video inpainting with pixmix, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 6, pp. 866–879, 2014.

[67] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, Image inpainting, in *Proc. SIGGRAPH*, 2000, pp. 417–424.

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

[69] I. Daribo and B. Pesquet-Popescu, Depth-aided image inpainting for novel view synthesis, in *Proc. MMSP*, 2010, pp. 167–170.

[70] J. Gautier, O. L. Meur, and C. Guillemot, Depth based image completion for view synthesis, in *Proc. 3DTV*, 2011, pp. 1–4.

[71] I. Ahn and C. Kim, A novel depth-based virtual view synthesis method for free viewpoint video, *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 614–626, 2013.

[72] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, Depth-aware patch-based image disocclusion for virtual view synthesis, in *Proc. SIGGRAPH Asia Technical Briefs*, 2015.

[73] C. Hao, Y. Chen, W. Wu, and E. Wu, Image completion with perspective constraint based on a single image, *Sci. China Inf. Sci.*, vol. 58, no. 9, pp. 1–12, 2015.

[74] N. Komodakis and G. Tziritas, Image completion using efficient belief propagation via priority scheduling and dynamic pruning, *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2649–2661, 2007.

[75] J. Habigt and K. Diepold, Image completion for view synthesis using markov random fields and efficient belief propagation, in *Proc. ICIP*, 2013, pp. 2131–2134.

[76] T. Ruzic and A. Pizurica, Context-aware patch-based image inpainting using Markov random field modelling, *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 444–456, 2014.

[77] B. Ceulemans, S.-P. Lu, G. Lafruit, and A. Munteanu, Efficient mrf-based disocclusion inpainting in multiview video, in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016.

[78] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, Patchmatch: A randomized correspondence algorithm for structural image editing, *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[79] K. He and J. Sun, Computing nearest-neighbor fields via propagation-assisted kd-trees, in *Proc. CVPR*, 2012, pp. 111–118.

[80] C. Barnes, F.-L. Zhang, L. Lou, X. Wu, and S.-M. Hu, Patchtable: Efficient patch queries for large datasets and applications, *ACM Trans. Graph.*, vol. 34, no. 4, p. 97, 2015.

[81] M. Bleyer, C. Rhemann, and C. Rother, Patchmatch stereo—stereo matching with slanted support windows, in *Proc. BMVC*, 2011.

[82] S. Gould and Y. Zhang, Patchmatchgraph: Building a graph of dense patch correspondences for label transfer, in *Proc. ECCV*, vol. 7576, 2012, pp. 439–452.

[83] B. Morse, J. Howard, S. Cohen, and B. Price, Patchmatch-based content completion of stereo image pairs, in *Proc. 3DIMPVT*, 2012, pp. 555–562.

[84] S.-P. Lu, J. Hanca, A. Munteanu, and P. Schelkens, Depth-based view synthesis using pixel-level image inpainting, in *Proc. DSP*, 2013, pp. 1–6.

[85] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, Performance optimizations for patchmatch-based pixel-level multiview inpainting, in *Proc. IC3D*, 2013, pp. 1–7.

[86] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, Depth image-based rendering with advanced texture synthesis for 3-D video, *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 453–465, 2011.

[87] P. Pérez, M. Gangnet, and A. Blake, Poisson image editing, *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.

[88] M. Kppel, X. Wang, D. Doshkov, T. Wiegand, and P. Ndjiki-Nya, Depth image-based rendering with spatio-temporal consistent textures synthesis for 3D video with global motion, in *Proc. ICIP*, 2012.

[89] G. Luo, Y. Zhu, Z. Li, and L. Zhang, A hole filling approach based on background reconstruction for view synthesis in 3D video, in *Proc. CVPR*, 2016, pp. 1781–1789.

[90] C. Li and M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in *Proc. CVPR*, 2016, pp. 2479–2486.

[91] H. Su, F. Wang, E. Yi, and L. J. Guibas, 3D-assisted feature synthesis for novel views of an object, in *Proc. ICCV*, 2015, pp. 2677–2685.

[92] Q. Huang, H. Wang, and V. Koltun, Single-view reconstruction via joint analysis of image and shape collections, *ACM Trans. Graph.*, vol. 34, no. 4, p. 87, 2015.

[93] M. Tatarchenko, A. Dosovitskiy, and T. Brox, Multi-view 3D models from single images with a convolutional network, in *Proc. ECCV*, 2016.

[94] Z. Zhu, H.-Z. Huang, Z.-P. Tan, K. Xu, and S.-M. Hu, Faithful completion of images of scenic landmarks using internet images, *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 8, pp. 1945–1958, 2015.

[95] A. Hornung, E. Dekkers, and L. Kobbelt, Character animation from 2d pictures and 3D motion data, *ACM Trans. Graph.*, vol. 26, no. 1, p. 1, 2007.

[96] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, Interactive images: Cuboid proxies for smart image manipulation, *ACM Trans. Graph.*, vol. 31, no. 4, p. 99, 2012.

[97] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, 3-sweep: Extracting editable objects from a single photo, *ACM Trans. Graph.*, vol. 32, no. 6, p. 195, 2013.

[98] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh, 3D object manipulation in a single photograph using stock 3D models, *ACM Trans. Graph.*, vol. 33, no. 4, p. 127, 2014.

[99] D. Hoiem, A. A. Efros, and M. Hebert, Automatic photo pop-up, *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.

[100] S.-P. Lu, S.-H. Zhang, J. Wei, S.-M. Hu, and R. R. Martin, Timeline editing of objects in video, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1218–1227, 2013.

[101] Y. Zhang, Y.-L. Tang, and K.-L. Cheng, Efficient video cutout by paint selection, *J. Comput. Sci. Technol.*, vol. 30, no. 3, pp. 467–477, 2015.

[102] F. L. Zhang, J. Wang, H. Zhao, R. R. Martin, and S. M. Hu, Simultaneous camera path optimization and distraction removal for improving amateur video, *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5982–5994, 2015.

[103] M. Lambooij, M. Fortuin, I. Heynderickx, and W. IJsselsteijn, Visual discomfort and visual fatigue of stereoscopic displays: A review, *J. Imaging Science and Technology*, vol. 53, no. 3, p. 30 201, 2009.

[104] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, and H.-P. Seidel, A perceptual model for disparity, *ACM Trans. Graph.*, vol. 30, no. 4, p. 96, 2011.

[105] P. Didyk, T. Ritschel, E. Eisemann, K. Myszkowski, H.-P. Seidel, and W. Matusik, A luminance-contrast-aware disparity model and applications, *ACM Trans. Graph.*, vol. 31, no. 6, p. 184, 2012.

[106] S.-P. Du, B. Masia, S.-M. Hu, and D. Gutierrez, A metric of visual comfort for stereoscopic motion, *ACM Trans. Graph.*, vol. 32, no. 6, p. 222, 2013.

[107] K. Templin, P. Didyk, K. Myszkowski, M. M. Hefeeda, H.-P. Seidel, and W. Matusik, Modeling and optimizing eye vergence response to stereoscopic cuts, *ACM Trans. Graph.*, vol. 33, no. 4, p. 145, 2014.

[108] T.-J. Mu, J.-J. Sun, R. R. Martin, and S.-M. Hu, A response time model for abrupt changes in binocular disparity, *The Visual Computer*, vol. 31, no. 5, pp. 675–687, 2015.

[109] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks, The zone of comfort: Predicting visual discomfort with stereo displays, *J. Vision*, vol. 11, no. 8, pp. 1–11, 2011.

[110] M. Lang, A. Hornung, O. Wang, S. Poulakos, A. Smolic, and M. Gross, Nonlinear disparity mapping for stereoscopic 3D, *ACM Trans. Graph.*, vol. 29, no. 4, p. 75, 2010.

[111] T. Basha, Y. Moses, and S. Avidan, Geometrically consistent stereo seam carving, in *Proc. ICCV*, 2011, pp. 1816–1823.

[112] K. Y. Lee, C. D. Chung, and Y. Y. Chuang, Scene warping: Layer-based stereoscopic image resizing, in *Proc. CVPR*, 2012, pp. 49–56.

[113] C. H. Chang, C. K. Liang, and Y. Y. Chuang, Content-aware display adaptation and interactive editing for stereoscopic images, *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 589–601, 2011.

[114] S.-J. Luo, Y.-T. Sun, I.-C. Shen, B.-Y. Chen, and Y.-Y. Chuang, Geometrically consistent stereoscopic image editing using patch-based synthesis, *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 1, pp. 56–67, 2015.

[115] W.-Y. Lo, J. van Baar, C. Knaus, M. Zwicker, and M. Gross, Stereoscopic 3D copy & paste, *ACM Trans. Graph.*, vol. 29, no. 6, p. 147, 2010.

[116] S.-J. Luo, I.-C. Shen, B.-Y. Chen, W.-H. Cheng, and Y.-Y. Chuang, Perspective-aware warping for seamless stereoscopic image cloning, *ACM Trans. Graph.*, vol. 31, no. 6, p. 182, 2012.

[117] Y. Niu, W.-C. Feng, and F. Liu, Enabling warping on stereoscopic images, *ACM Trans. Graph.*, vol. 31, no. 6, p. 183, 2012.

[118] R.-F. Tong, Y. Zhang, and K.-L. Cheng, Stereopasting: Interactive composition in stereoscopic images, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 8, pp. 1375–1385, 2013.

[119] S.-P. Du, S.-M. Hu, and R. R. Martin, Changing perspective in stereoscopic images, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 8, pp. 1288–1297, 2013.

[120] T.-J. Mu, J.-H. Wang, S.-P. Du, and S.-M. Hu, Stereoscopic image completion and depth recovery, *The Visual Computer*, vol. 30, no. 6, pp. 833–843, 2014.

[121] S. Avidan and A. Shamir, Seam carving for content-aware image resizing, *ACM Trans. Graph.*, vol. 26, no. 3, p. 10, 2007.

[122] F. Raimbault and A. Kokaram, Stereo video inpainting, *Proc. SPIE*, vol. 7863, p. 78631, 2011.

[123] F. Liu, Y. Niu, and H. Jin, Joint subspace stabilization for stereoscopic video, in *Proc. ICCV*, 2013, pp. 73–80.

[124] M. Irani, Multi-frame correspondence estimation using subspace constraints, *Int. J. Comput. Vision*, vol. 48, no. 3, pp. 173–194, 2002.

[125] S. Kopf, B. Guthier, C. Hipp, J. Kiess, and W. Effelsberg, Warping-based video retargeting for stereoscopic video, in *Proc. ICIP*, 2014, pp. 2898–2902.

[126] Y. Liu, L. Sun, and S. Yang, A retargeting method for stereoscopic 3D video, *Comp. Visual Media*, vol. 1, no. 2, pp. 119–127, 2015.

[127] M. Wang, X.-J. Zhang, J.-B. Liang, S.-H. Zhang, and R. R. Martin, Comfort-driven disparity adjustment for stereoscopic video, *Comp. Visual Media*, vol. 2, no. 1, pp. 3–17, 2016.

[128] N. Jiang, P. Tan, and L.-F. Cheong, Multi-view repetitive structure detection, in *Proc. ICCV*, 2011, pp. 535–542.

[129] P. Y. Laffont, A. Bousseau, and G. Drettakis, Rich intrinsic image decomposition of outdoor scenes from multiple views, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 2, pp. 210–224, 2013.

[130] A. Djelouah, J.-S. Franco, E. Boyer, F. L. Clerc, and P. Prez, Multi-view object segmentation in space and time, in *Proc. ICCV*, 2013, pp. 2640–2647.

[131] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z.-

H. Zhou, Multi-view video summarization, *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, 2010.

[132] I. A. Shamir, H. S. Park, Y. Sheikh, J. K. Hodgins, and Ariel, Automatic editing of footage from multiple social cameras, *ACM Trans. Graph.*, vol. 33, no. 4, p. 81, 2014.

[133] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung, Videosnapping: Interactive synchronization of multiple videos, *ACM Trans. Graph.*, vol. 33, no. 4, p. 77, 2014.

[134] T. Nguyen, G. Reitmayr, and D. Schmalstieg, Structural modeling from depth images, *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 11, pp. 1230–1240, 2015.

[135] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, 3D semantic parsing of large-scale indoor spaces, in *Proc. CVPR*, 2016, pp. 1534–1543.

[136] S.-P. Lu and S.-H. Zhang, Saliency-based fidelity adaptation preprocessing for video coding, *J. Comput. Sci. Tech.*, vol. 26, no. 1, pp. 195–202, 2011.

[137] R. Florea, A. Munteanu, S.-P. Lu, and P. Schelkens, Wavelet-based L_∞ semi-regular mesh coding, *IEEE Trans. Multimedia*, 2016. doi: 10.1109/TMM.2016.2614483.

Shaoping Lu is a senior researcher in Department of Electronics and Informatics (ETRO) at Vrije Universiteit Brussels (VUB), where he is currently leading a new team focusing on 3-D video processing. He received the PhD degree in computer science from Tsinghua University, China, in 2012. From Nov. 2012, he has been a postdoc at VUB. His primary research areas are image and video processing, with particular interests in multiview video acquisition, representation, compression, and rendering.

Taijiang Mu is currently a postdoctoral researcher in the Department of Computer Science and Technology, Tsinghua University, where he received the PhD and BS degrees in 2016 and 2011, respectively. His research interests include computer graphics, stereoscopic image/video processing, and stereoscopic perception.

Songhai Zhang received the PhD degree in computer science in 2007 from Tsinghua University, Beijing. He is currently an associate professor in the Department of Computer Science and Technology of Tsinghua University, Beijing. His research interests include image and video processing and geometric computing.