Image- and Video-based Rendering

Christian Lipski, Anna Hilsmann, Carsten Dachsbacher and Martin Eisemann

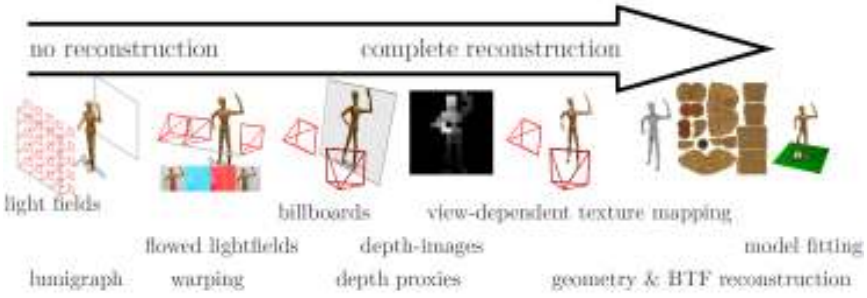

Figure 17.1: Overview of video and image-based rendering systems. While some approaches are based on densely sampling scene appearance with many images (far left), others rely on having available high-quality 3D scene geometry (far right). Numerous techniques can be located somewhere between these two limiting cases.

17.1 Introduction

The purpose of image- and video-based rendering (IVBR) is to be able to synthesize photo-realistic new, virtual views of real-world scenes and events from no more than a set of conventional photographs or videos of the scene. Purely image-based, or plenoptic, approaches densely sample scene appearance using a large number of images, Sections 17.2 and 5.3. New views are generated by simply re-sampling the captured image data. In contrast, geometry-assisted methods require much less input images. Here, 3D scene geometry is either a-priori known, reconstructed from the captured imagery, or acquired separately by some other means, e.g., ranging imaging, Chapter 4. With (approximate) scene geometry available, views

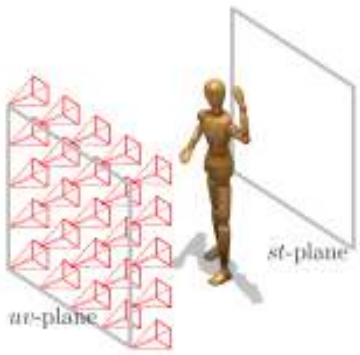

Figure 17.2: Light field acquisition: in the two-parallel-plane parameterization, the plenoptic function of a scene is regularly sampled by camera positions on the uv plane and coinciding image plane st. To acquire the light field of a static scene, a motorized camera gantry can be used, Section 5.2, while for dynamic scenes an array of video cameras is needed.

from arbitrary viewpoints are synthesized using the acquired images as geometry texture, Section 17.3. Over the years, various IVBR methods have been proposed that can all be categorized in-between these two limiting cases [Lengyel 98], Fig. 17.1. Their respective advantages and limitations have been discussed in several IVBR surveys [Shum and Kang 00, Smolic et al. 09, Linz 11, Germann 12].

17.2 Plenoptic Approaches

Light Fields and Lumigraphs

The idea underlying light field rendering is to represent the plenoptic function of a real-world scene, i.e., its appearance from any direction, as a four-dimensional lookup table [Levoy and Hanrahan 96], Fig. 17.3. By assuming the space around the scene to be transparent, each light ray can be parameterized by four scalar values (u, v, s, t). (u, v) are the intersection coordinates of the light ray with the camera plane, while (s, t) are the ray's intersection coordinates with the fronto-parallel image plane. Light field acquisition consists of sampling the uv plane by taking images from regularly spaced uv grid positions, Fig. 17.2. For high-quality light field acquistion, a rectifying homography transformation is typically applied to align the image plane of all captured images with the st plane [Kim et al. 13].

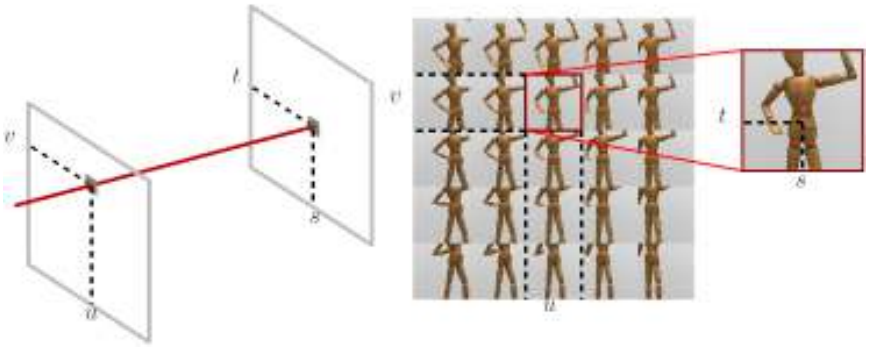

Figure 17.3: Light field rendering: views of the scene from arbitrary vantage points are obtained by tracing a view ray for each pixel and determining the intersection coordinates with the uv- and the st-planes (left). The $uvst$ coordinates are used to look up pixel color from the 4D light field data set (right).

For static scenes, a mechanical gantry enables sequentially capturing many light field images with one camera. The capture hardware has to be precisely calibrated, however, and the capture process may take a long time. Alternatively, single-chip light field cameras have been propsed [Rodriguez-Ramos et al. 11, Lytro, Inc. 12, Wietzke 12] that employ lenslet arrays to acquire the entire light field simultaneously, Chapter 5. With single-chip systems, however, there is a trade-off between image resolution (st plane) and viewpoint range (uv plane).

Several modifications and extensions to light field rendering have been proposed. In Lumigraph rendering, capturing the scene is simplified by allowing for non-regular placement of cameras [Gortler et al. 96]. Before rendering, the image data is re-parameterized to the (u, v, s, t) representation in a rebinning step. Following a similar approach, light fields can also be captured using mobile phones [Davis et al. 12].

Light field rendering requires neither scene geometry nor image correspondence information. In theory, photo-realistic, high-qualty rendering results can be obtained. For aliasing-free rendering, however, unrealistically high sampling rates are required [Chai et al. 00]. To reduce discretization artifacts when light field-rendering from undersampled data, filtering schemes can be employed [Stewart et al. 03, Eisemann et al. 07]. On the other hand, light field data is highly redundant which allows for efficient compression, storage, and streaming [Vaish and Adams 12].

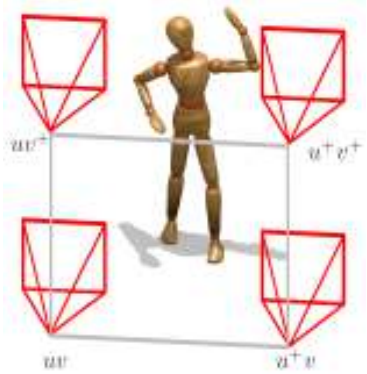

Figure 17.4: Sparse light field capture: with increasing distance between camera positions, uv plane sampling becomes less dense. To avoid ghosting artifacts during rendering, disparity between light field images must be compensated.

Flowed Light Field Rendering

To render from only sparsely sampled light fields, image warping can be employed to synthesize in-between camera positions on the uv-plane [Shade et al. 98,Heidrich et al. 99,Einarsson et al. 06], Fig. 17.4. Prior to rendering, dense image correspondences are established between adjacent light field images by estimating the optical flow fields. During rendering, each ray is intersected with the uv-plane, but instead of just looking up pixel color in the closest-by light field image or linear blending, backward warping is applied to correct for parallax. In backward warping, the pre-computed flow fields are first individually forward-warped to the desired location [Shade et al. 98]. For each ray intersection with the uv plane, the flow vectors to all input image are then looked up and the corresponding light field image pixels are weightedly blended, Fig. 17.5.

Flowed light field rendering requires considerably less input images than standard light field rendering. For a full 360° surround capture of an actor, 3×30 images are sufficient to obtain convincing rendering results [Einarsson et al. 06]. In addition, the warping step can correct for small errors, e.g., misaligned images due to calibration errors or unsynchronized cameras. This way, even dynamic scenes may be light field-captured sequentially from different vantage points [Einarsson et al. 06].

The main limitation of flowed light field rendering is its dependency on correct, dense flow fields. With increasing distance between neighboring camera positions or for complex scenes, however, optical flow estimation algorithms tend to fail. Another limitation is that backward warping can-

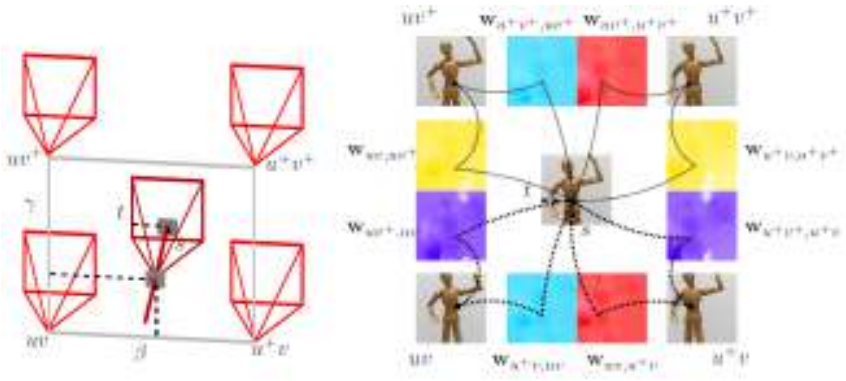

Figure 17.5: Flowed light field rendering. For each viewing ray that intersects the uv plane, the four surrounding images $u[^+]v[^+]$ are determined. In a pre-processing step, optical flow fields $\mathbf{w}_{u[^+]v[^+],u[^+]v[^+]}$ have been computed between adjacent light field images. During rendering, backward warping $\mathbf{w}$ is applied to obtain the ray's corresponding pixel color in each of the four images. The rendered pixel is then computed as a weighted sum from the four light field images.

not cope with object occlusions, causing ghosting and other artifacts during rendering.

Warping-Based Approaches

In contrast to flowed light field rendering where for each viewing ray adjacent light field images are locally queried, in warping-based rendering, also known as image morphing or correspondence-based rendering, the acquired light field images are being warped completely using flow fields, Fig. 17.6. Warping-based approaches are not restricted to light fields but have a long-standing tradition in view interpolation and the creation of smooth transitions between similar images [Stich et al. 08]. For example, warping has been used to create transitions between different actors who are performing an identical choreography [Beier and Neely 92]. An extension to more than two images has been proposed by Lee et al. [Lee et al. 98]. Chen and Williams [Chen and Williams 93] proposed to use forward image warping for viewpoint interpolation based on previously estimated flow fields. Image warping is applied to each input image, and the final result is obtained by weighted blending of the warped images. The combination of image warping and blending is also frequently referred to as image morphing.

For real-world images, perceptually convincing dense correspondence

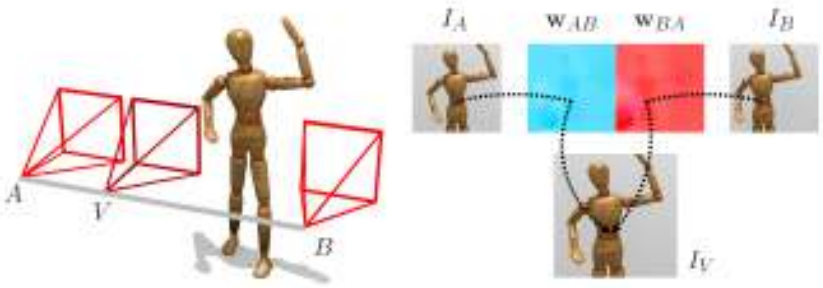

Figure 17.6: Warping-based rendering: to interpolate viewpoint I_V, each pixel in the input images I_A and I_B is foward-warped according to its flow vector $\mathbf{w}_{AB}$, $\mathbf{w}_{BA}$. The final image is weightedly blended from the two warped images.

fields can be estimated either automatically [Stich et al. 11] or assisted by additional user input [Ruhl et al. 12a]. For multi-view video footage of dynamic scenes, additionally loop consistency among subsequent video frames from neighboring cameras can be exploited [Sellent et al. 12]. For synthetic scenes, the flow vector for a given pixel location $\mathbf{x}$ can be easily derived from per-pixel depth d and camera matrices $\mathbf{P}_A, \mathbf{P}_B$:

$$\mathbf{w}_{AB}(\mathbf{x}) = \mathbf{P}_B(\mathbf{P}_A{}^{-1}(\mathbf{x}, \mathbf{d})) \tag{17.1}$$

where $\mathbf{P}_A{}^{-1}$ is the inverted projection matrix of camera A, Section 1.6. To ensure geometrically undistorted in-between views during warping-based image interpolation, rectifying homography transforms are applied to the input impages prior to warping [Seitz and Dyer 96]. Warping enables convincing viewpoint interpolation of panoramas [McMillan and Bishop 95] (Chapter 3), dynamic light fields [Goldlücke et al. 02], uncalibrated images [Fusiello 07] as well as uncalibrated and unsynchronized multi-view video [Lipski et al. 10a].

Similarities exist to flowed light fields as well as to depth-based rendering, Section 17.3. If backward warping is used, i.e., if each target pixel of the view to be synthesized is queried for its location in the captured light field images, image warping is a special case of flowed light fields: here, the interpolated viewpoint is located on the manifold spanned by all camera positions, as opposed to flowed light fields where the target view does not have to lie on the capture manifold. On the other hand, if forward-warping is used, i.e., if each pixel of a captured light field image gets shifted to its new location in the target view, warping is akin to depth-based rendering: For rectified input imagery, the flow vectors are scanline-aligned and reduce

to one-dimensional *disparity* that is proportional to the inverse scene depth at a given pixel position.

Analogous to flowed light fields, warping-based rendering requires considerably less input images than pure light field rendering. The amount of image data can be further reduced if cameras do not have to span a 2D manifold but can be arranged in arc-like set-ups around the scene. Warping is able to compensate for calibration inaccuracies and works with unsynchronized multi-view video footage. It performs robustly even for complex outdoor scenes [Lipski et al. 10a] and lends itself creating numerous space-time visual effects [Linz et al. 10b].

In comparison to traditional light field rendering, one limitation of warping-based IVBR is that the virtual viewpoint must lie on the manifold spanned by the capture positions of all input images/videos. The quality of the rendered output depends on the accuracy of the dense correspondence maps. For multi-view acquisition setups of up to about 10° between adjacent cameras, robust correspondence estimation algorithms exist [Lipski et al. 12], Chapter 8. Still, occlusion effects cannot always be handled correctly by warping alone, motivating the use of geometry proxies in IVBR.

17.3 Geometry-Assisted Approaches

Geometry Proxies

Image-based occlusion detection is an active research area [Ince and Konrad 08, Herbst et al. 09]. Alternatively, actual depth information is needed to render occlusion effects robustly [Chen and Williams 93]. Fortunately, even if actual scene geometry is too complex for faithful reconstruction, simple geometry proxies often already suffice to achieve visually convincing rendering results.

The most basic scene geometry approximation consists of a single fronto-parallel plane located in the middle of the scene, often referred to as a billboard, Fig. 17.7. By default, the billboard is always oriented perpendicular to the current viewing direction. For rendering, one or more captured scene images are projected onto the billboard and rendered as textures, cross-blending between projected images [Snavely et al. 06, Snavely et al. 08a]. If the scene consists of several objects, separate billboards may be used, one for each object. An individual object may also be represented also by more than one billboard. In microfacet billboarding [Yamazaki et al. 02], the object is divided into many thousand small billboards. Further rendering improvements can be achieved by faithfully reconstructing the boundary colors of neighboring proxies [Germann et al. 10], by merg-

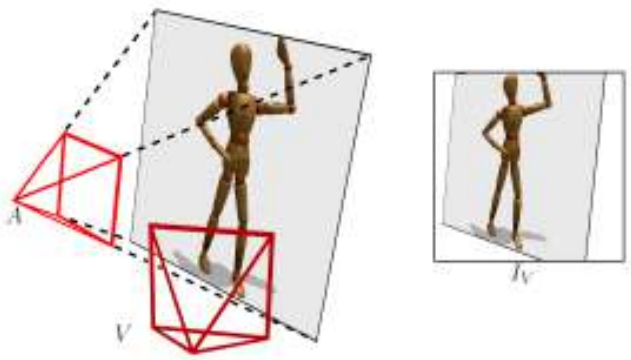

Figure 17.7: In geometry proxy-based rendering, only a coarse geometric representation of the scene may be required. In the depicted case, a single view-dependent, fronto-parallel billboard is used. During rendering, the source image is projected onto the proxy geometry.

ing them in image space [Hornung and Kobbelt 09], or by applying local displacements to billboards [Waschbüsch et al. 07]. The main advantage of billboarding is its modest computational requirements, enabling real-time, on-the-fly rendering from live-captured multi-video footage [Goldlücke and Magnor 03]. Billboard rendering has been successfully applied to free-viewpoint video of actors, sports broadcasts, and architectural scenes [Germann et al. 10, Schwartz et al. 10].

If for some image region no reliable billboard depth can be obtained, to avoid more annoying artifacts such regions may deliberately be visualized in a blurred manner, Section 17.4. In ambient point cloud rendering, for example, random depth values are assigned to pixels of unknown depth so that they form an amorphous, unobtrusive point cloud [Goesele et al. 10].

Geometry proxy-based approaches are able to achieve visually pleasing results without pixel-accurate depth information. Coarse scene representations such as billboards can be estimated very robustly, and computational as well as memory requirements are small due to the limited amount of estimated geometric information. Billboard rendering yields improved rendering results from undersampled light field data. Also, viewpoint position is not restricted to any kind of camera manifold. Still, rendering artifacts may be apparent, Section 17.4. Especially when using a simple billboard proxy, strong ghosting artifacts become visible when cross-blending from widely separated input images. The impact of such artifacts in image-based rendering on perceived image quality has been studied by Vangorp et al. [Vangorp et al. 11]. Another constraint is that although only coarse

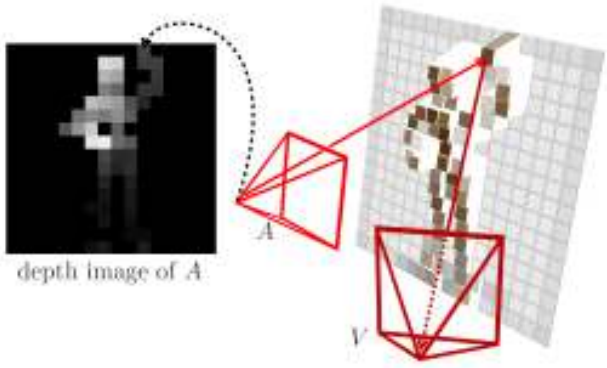

Figure 17.8: In depth-image-based rendering (DIBR), the depth of each pixel is known (visualized by greyscale depth map, left). According to pixel depth and camera matrix of camera image I_A, each pixel is projected to its world space position. Using the projection matrix of the virtual camera I_V, the image I_A is projected to the image plane of I_V.

depth information is required, the input images must be calibrated.

Depth Image-Based Rendering (DIBR)

Depth image-based rendering relies on dense depth information for every pixel of all input images. To obtain reliable depth, dense per-pixel stereo matching algorithms are typically used, Chapter 8. For rendering, in general each pixel in reference image I_A is reprojected into the world space and re-rendered from the desired viewpoint I_V [Fehn 04], Fig. 17.8. In point cloud rendering [Zabih and Woodfill 94, Addison et al. 95] and point splatting [Hornung and Kobbelt 09], reprojection and re-rendering is accomplished by treating all pixels independently. Alternatively, the source images can be considered as a connected mesh [Zitnick et al. 04, Zheng et al. 09]. To avoid artifacts along object silhouettes, single quads of the mesh that feature large depth discontinuities must be locally discarded. Alpha matting is used to estimate local foreground color and alpha values, and an additional boundary layer is rendered to guarantee smooth transitions between different depth layers.

A very challenging problem is disocclusion handling. Both point splatting and mesh rendering methods may produce holes in the final image [Tauber et al. 07]. One solution is to use two or more source images for rendering that hopefully fill in the holes in the final image [Zitnick et al. 04, Zheng et al. 09]. Another possibility is to use a depth-layered

representation of the scene [Shade et al. 98, Müller et al. 08]. Still, infilled image regions cannot be ruled out and may remain annoyingly visible as empty holes in the synthetized novel view. To remedy the problem, several inpainting techniques have been proposed to assign plausible color information to such unfilled regions [Criminisi et al. 03, Moreno-Noguer et al. 07, Debevec et al. 98]. Common inpainting techniques have also been surveyed and benchmarked for their applicability in image-based rendering [Schmeing and Jiang 11].

If dense and correct depth information can be obtained, depth image-based rendering can be very accurate. Similar to depth proxies, the location of the virtual viewpoint is arbitrary. Also, only few input views are needed to achieve useful rendering results. On the other side, accurate, dense depth reconstruction is notoriously difficult and error-prone for the general case. Acquisition cameras must be calibrated and synchronized accurately, and scene content may not change appearance by too much between different viewpoints, neither due to occlusion effects nor to non-Lambertian reflectance characteristics. In essence, input images may not be separated by more than about 10°, else even state-of-the-art reconstruction methods fail, Chapter 8.

To overcome the problems specific to depth-based and warping-based rendering, a hybrid approach has been proposed [Lipski et al. 14]. By exploiting both dense, pairwise image correspondences as well as depth information simultaneously, convincing rendering results can be obtained even from imprecisely reconstructed depth and inaccurately calibrated, asynchronously captured multi-view footage.

3D Geometry Reconstruction and View-Dependent Texture Mapping

If the scene is not too complex, instead of estimating local per-pixel depth it may be possible to reconstruct a complete, consistent 3D geometry model of the scene prior to rendering, Chapter 12. During rendering, the 3D mesh can then be projectively textured using a view-dependent selection of input images [Debevec et al. 98], Fig. 17.9. If the scene consists of a single object of interest, visual hulls are a common, conservative 3D geometry approximation [Baumgart 74, Potmesil 87, Matusik et al. 00]. While visual hull reconstruction and rendering is very fast and allows for real-time applications [Li et al. 03], due to its inherently limited geometric accuracy attainable rendering quality is limited. Instead, more elaborate, off-line 3D reconstruction schemes can be employed. One approach consists of performing structure-from-motion calibration [Snavely et al. 06], Chapter 7, prior to applying quasi-dense multi-view reconstruction of surface patches [Fu-

rukawa and Ponce 10, Snavely 12, Lipski 12], Chapter 8. Alternatively, depth cameras or 3D scanners may be used to obtain partial object geometry [Wood et al. 00], Chapter 9. Finally, a watertight 3D model is obtained from the acquired point cloud using Poisson surface reconstruction [Kazhdan et al. 06], Chapter 10. Solutions to estimate 3D geometry exist also for scenes that contain non-Lambertian objects [Vogiatzis et al. 06]. Many reconstruction algorithms rely on silhouette extraction, effectively limiting applicability to scenes consisting of a single, easily segmentable object of interest. If automatic reconstruction is infeasible, scene geometry reconstruction can also be user-guided, e.g., by specifying points and edges of the mesh to determine 3D-positions semi-automatically [van den Hengel et al. 07]. Because of their relevance for popular 3D map services, specific tools have been developed for architectural scenes that allow for user assistance and correction [Debevec et al. 96, google 12]. User-assisted methods are, however, labor-intensive.

For view-dependent texture mapping, only a small number of input views is needed to reproduce highly realistic scene appearance. Given an accurate 3D model, the scene can be rendered from any viewpoint, and the rendered viewpoint is not restricted to any particular area. Having a 3D geometry model available has additional advantages, e.g., it can receive and cast shadows in a virtual scene. On the other hand, exact alignment between projected images and 3D geometry is essential for authentic rendering results. If camera calibration is only slightly off, or if the geometry model exhibits even small inaccuracies, annoying rendering artifacts occur [Eisemann et al. 08].

Model-based IVBR

3D geometry can be reconstructed from multi-view imagery based on first principles, Chapter 8. However, scene reconstruction can be considerably improved if knowledge about scene content is exploited and a parameterized 3D model can be provided. By fitting an a-priori 3D model to the recorded data, parameter space is greatly reduced and consistency enforced, Chapter 12. Prominent application areas are human pose estimation and motion capture, Chapter 11. For free-viewpoint video rendering of an actor, for example, joint angles and shape parameters of a 3D human body model as well as time-varying textures may be derived directly from sparse multi-video footage [Carranza et al. 03]. A statistical human body model enables modeling almost any person's physique [Hasler et al. 09a], even from a single video recording [Jain et al. 10]. Alternatively, laser scanning or other depth sensors can be employed to model an individual's 3D geometry precisely [de Aguiar et al. 08b, Ye et al. 11, Kuster et al. 11]. Of course, known 3D geometry and appearance of scene background or other parts of the scene

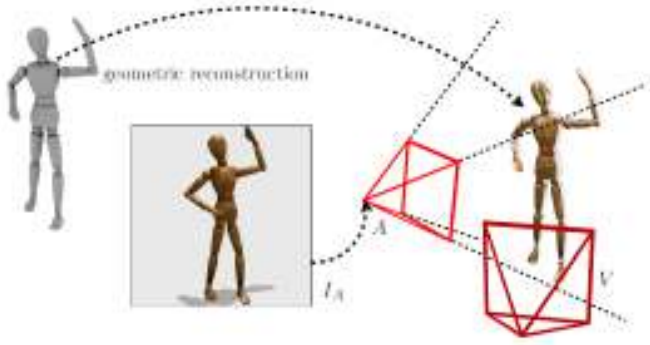

Figure 17.9: View-dependent texture mapping: if an accurate 3D geometry model of the scene is available, it can be projectively textured using only a few captured input images. Typically, more than one input image is used for projective texturing to cover all visible regions of the object. The selection of input images and their respective blending weights are assigned based on the position of the viewpoint.

Figure 17.10: Model-based IVBR: a-priori information about scene content can be exploited to obtain robust modeling results. Instead of reconstructing 3D geometry from scratch, a parameterized 3D model may be fitted to the recorded footage.

can also be exploited. Many spectator sports, for example, take place on a well-defined playing field. In TV sports broadcasting applications, Chapter 21, this a-priori knowledge is used for reliable background segmentation as well as scene augmentation [Hilton et al. 11, Germann et al. 10].

Dynamic Objects and Scenes

Many image-based approaches do not explicitly provide any special means to deal with dynamic scenes recorded with multiple video cameras simultaneously. Although it is always possible to apply image-based techniques independently to each consecutive frames of a multi-video sequence, this is no way to assure temporal coherence. Consequently, independent processing of consecutive time frames can result in flickering artifacts, severely impacting perceived rendering quality. In contrast, video-based rendering approaches are specifically designed for dynamic scene content by both ensuring and exploiting temporal coherence [Magnor 05].

If multi-video acquisition is synchronized across all cameras, an initial geometry model may be estimated for the first frame [Furukawa and Ponce 10] and tracked over successive frames [Furukawa and Ponce 08]. Occlusion of scene parts, however, can lead to holes in the surface model. Mesh completion may be employed to obtain a temporally coherent representation [Li et al. 12a]. For deformable objects like the human face, an initial mesh may be tracked using one or several reconstruction anchor frames [Bradley et al. 10, Beeler et al. 11].

Alternatively, instead of reconstructing a geometry model first and enforcing its temporal consistency in a second step, dynamic geometry may be reconstructed globally as a weighted minimal 3D hyper-surface in 4D space-time [Goldluecke and Magnor 04]. The hyper-surface is defined by the minimum of an energy functional which is given by an integral over the entire hypersurface and which is designed to optimize photo-consistency. A PDE-based evolution derived from the Euler-Lagrange equation maximizes consistency with all of the given video data simultaneously. The result is a globally photo-consistent, closed 3D model of the scene that varies smoothly over time [Goldlücke and Magnor 04].

Besides surface geometry, some IVBR applications also benefit from dense 3D scene motion. This so-called scene flow can be reconstructed from synchronized multi-video footage by estimating the optical flow per camera view and reprojecting the flow fields onto 3D scene geometry [Vedula et al. 05].

While synchronized multi-video recordings considerably simplify subsequent processing, mass-market cameras typically do not provide any technical means for inter-camera synchronization [Hasler et al. 09a]. Even with high-end camera equipment, temporally misaligned frames and complete frame drops can occur [Imre and Hilton 12, Imre et al. 12]. Only a few IVBR approaches explicitly allow for non-synchronized multi-view input imagery. One option is to synchronize dynamic light field recordings prior to rendering via temporal interpolation [Wang et al. 07]. By deliberately offsetting camera recording times, temporal resolution can even be im-

proved [Li et al. 12b]. For the initial spatial reconstruction, however, still a subsets of cameras has to record the dynamic scene in sync.

Besides the technical or practical inability to synchronize multiple video cameras, e.g., in the field, also multi-camera calibration can be a tedious, difficult, and error-prone procedure, Section 2.3. A warping-based approach still enables free-viewpoint video of complex outdoor scenes from completely unsynchronized, uncalibrated, sparse multi-video footage [Lipski et al. 10a]. The underlying idea is to simulate a virtual video camera by interpolating between recorded video frames across space and time. Prior to rendering, dense image correspondences must be estimated between consecutive video frames of each camera sequence as well as between adjacent cameras [Stich et al. 11]. View interplation takes place in the spatio-temporal domain spanned by all recorded video frames [Stich et al. 08]. By subdividing the interpolation domain into tetrahedrons, with the recorded video frames as vertices and dense correspondence maps along the edges, free viewpoint navigation, slow motion, freeze-and-rotate shots, and many more special effects can be photo-realistically rendered [Linz et al. 10b].

17.4 Advanced Image-based Methods and Extensions

The classification introduced in the previous sections gives an overview of fundamental IVBR approaches. When looking at some actual rendering systems it is apparent that many of them do not fit precisely into one single category. Some approaches have been proposed that combine different techniques. Unstructured lumigraphs [Buehler et al. 01], for example, generalize both lumigraphs and view-dependent texture maps. Depending on the level of detail of the proxy geometry, they behave like one of the both extremes, or a mixture of them. The view-dependent texture mapping approach [Debevec et al. 96] also employs depth-based rendering at a fine level. For each reconstructed facade, the original textures can be projected onto the geometry, and local depth maps are computed to compensate local projection errors. View-dependent textured splatting [Yang et al. 06c], on the other hand, constitutes a mixture between view-dependent texture mapping and point splatting.

For many practical applications it proves to be beneficial to segment the scene into different regions (e.g., actors and background) and to treat them differently. In outdoor sports scenarios, for example, players are separated from the field at an early stage of the processing pipeline [Hilton et al. 11, Germann et al. 10]. While billboard representations or 3D surfaces are reconstructed for the individual players, it is often sufficient to represent the playing field by a single plane. Alternatively, user-supervised

segmentation and billboard rendering is used for the foreground person while the background is reconstructed in high detail and rendered using view-dependent texture mapping [Ballan et al. 10].

Error-Concealed Rendering

As each image-based method has its own advantages, it also has its particular limitations and failure cases. In some scenarios, the user can assist the reconstruction process to obtain pleasing results. Several approaches have been suggested that require user input for geometry reconstruction [Debevec et al. 96, van den Hengel et al. 07]. Other approaches require sparse user input for scene segmentation [Ballan et al. 10, Guillemaut et al. 10] and view interpolation [Chaurasia et al. 11]. Floating textures provide an automatic correction mechanism at render time that does not require any manual intervention [Eisemann et al. 08]. Prior to the blending stage in image-based rendering, the different source image projections on the geometry proxy are locally aligned based on optical flow estimated in real-time. Alternatively, the 3D geometry may be aligned with the images using sparse feature matches, circumventing dense optical flow estimation [Germann et al. 12].

Comprehensive Reconstruction

For free-viewpoint video, i.e., rendering a dynamic, real-world scene from arbitrary vantage points, 3D scene geometry must be available in some form. To allow also for illumination changes of the scene or for augmentation, in addition surface reflectance properties must be estimated. If the scene is diffusely reflecting, the reconstruction of one consistent texture map suffices. Different approaches exist to estimate a consistent diffuse texture atlas from multi-view imagery given 3D geometry, illumination, and camera parameters [Wang et al. 01, Lempitsky and Ivanov 07, Gal et al. 10]. 3D geometry and consistent texture may also be estimated simultaneously [Matsuyama et al. 04, Starck and Hilton 07b, Liu et al. 10b, Schwartz et al. 11b, Autodesk 12, Agisoft 12, Hypr3D Development Team 12, Nguyen et al. 12].

In constrast to Lambertian objects, comprehensive reconstruction of scenes containing specular, glossy, semi-transparent, or mirroring surfaces is considerably more difficult [Ihrke et al. 10]. For non-Lambertian surfaces, the ability to recover normal directions accurately varies varies greatly with both actual surface BRDF and illumination pattern and may even be ill-posed [DŹmura 91]. If 3D scene geometry is known, inverse rendering allows recovering illumination and/or BRDF, represented in spherical harmonics basis functions [Ramamoorthi and Hanrahan 01a]. Alternatively, parame-

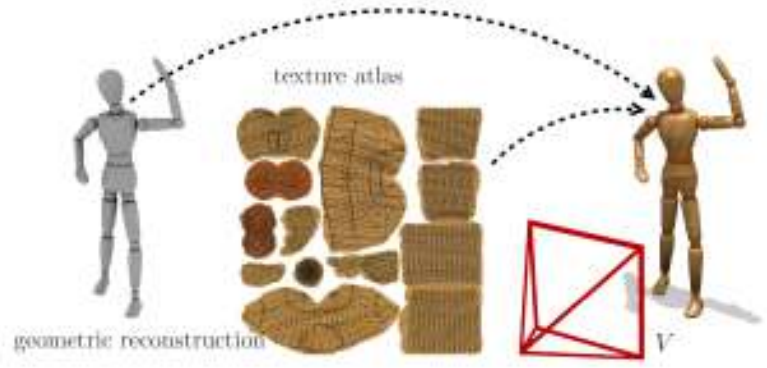

Figure 17.11: Comprehensive reconstruction: if both surface geometry and reflectance properties of the scene can be modeled from the input imagery, the traditional 3D rendering pipeline can be utilized to render the scene from any viewpoint and under arbitrary illumination.

terized reflection models may be fitted to match captured multi-view scene appearance, either for known [Theobalt et al. 07] or unknown [Li et al. 13] scene illumination. If scene appearance from only a single viewpoint is to be varied for different illumination conditions, high-speed recording under time-multiplexed illumination allows relighting the scene [Wenger et al. 05].

Image-based Object and Scene Manipulation

Image- and video-based rendering approaches concentrate on view interpolation. In recent times, image-based methods have been proposed also for modeling and rendering appearance variations. A set of parameters is used to span a domain to search and warp the images of an object. For articulated objects, for example, such a parameter set can be defined by a skeletal pose representation [Xu et al. 11, Hilsmann et al. 13], Chapter 11, or by silhouette shape features [Hauswiesner et al. 13]. For facial expressions, facial feature locations in the image can be used [Zhang et al. 06]. During synthesis, a temporal coherent matching strategy is used to identify the *nearest* database image(s) to the given configuration of pose or facial expression in descriptor space. The retrieved images are then used to synthesize an image for this pose configuration. The best matching database image is mapped onto an animated 3D model of a human. As the retrieved image might not show exactly the same pose as required, fine-scale warping of the rendered image is necessary. Other approaches synthesize articulated human body poses by a convex combination of example poses [Casas et al. 14].

The interpolation domain for skeletal poses or facial expressions is much

more complex and of higher dimensionality than the interpolation domain necessary for view synthesis. Possible expressions/poses that can be synthesized from the database are restricted to the convex hull of examples, and sampling the space such that every possible expression/pose can be synthesized becomes intractable. This limitation has been addressed by splitting up the object, i.e., the face or human body, into subregions that are assumed to be more or less independent. Each of these regions then has its own descriptor space and is synthesized from different examples images. To produce the final image, the subregion images are seamlessly blended.

Compositing, Augmentation, and Consolidation with Traditional Computer Graphics

The paramount goal of image- and video-based rendering techniques is to capture real environments and synthesize novel views. However, if further interaction with or editing of the scene is required several additional problems occur, many of them still unsolved. One challenge is realistic compositing of image- and video-based rendering results with those from traditional 3D rendering approaches. It is as apparent that there are many applications where real content is to be complemented with synthetic assets, or the acquired 3D data is (partly) used in otherwise synthetic scenes. For example, few blockbuster movies nowadays are not augmented with 3D renderings in the form of special effects, Chapter 20. Also live TV sports broadcasts and other application areas rely on convincingly augmenting real-world footage with synthetic 3D-rendered content, Chapters 21, 23.

Arguably the simplest solution to combining image- or video-based rendering with traditional 3D rendering is to 3D-reconstruct the scene from the recorded video footage, Part II, augment the scene it in 3D world space, and 3D-render it again. However, image-based 3D reconstruction methods have numerous limitations. A reconstruction method may be applicable only to single objects and may require a controlled environment for capture, or recorded scene appearance is not factorized into lighting and reflectance preventing photo-realistic augmentation with other 3D models, or the scene is optically too complex for faithful reconstruction. These restrictions give rise to several interesting research challenges towards realistic augmentation of real-world scenes with synthetic 3D graphics. Among the different aspects that need to be considered are correctly matching the illumination of the virtual object to that of the real-world scene as well as color bleeding and shadows cast by the virtual object onto the real-world scene, and vice versa. Fortunately, the task of augmenting a complex, dynamic real-world scene with virtual objects can, in essence, be reduced to consistently augmenting each video frame separately.

Illumination Reconstruction The appearance of a synthetic object is determined by its 3D shape, surface reflectance characteristics, and illumination. If the goal is to augment some virtual object into an image, its 3D shape and reflectance properties are known. Only the lighting conditions of the scene in the image are unknown and must be reconstructed. One simplification that is often made is that only the far-field illumination of the scene is reconstructed, i.e., the scene is assumed to be illuminated by a hemisphere infinitely far away so the illuminating light distribution can be represented as an environment map. Scene illumination can be acquired directly by placing a so-called light probe, often a mirroring sphere, into the real-world scene and taking a photo of it [Debevec 98]. Alternatively, a fish-eye lens can be used [Sato et al. 99]. If the recorded scene is not accessible anymore, illumination may be interactively estimated using some (coarse) scene geometry proxy [Karsch et al. 11]. In many cases, illumination estimation is tightly coupled with the reconstruction of overall scene appearance in a joint optimization approach [Kholgade et al. 14, Rogge et al. 14, Hara et al. 08, Haber et al. 09]. Regularly, surface reflectance of the real-world scene is assumed to be Lambertian [Ramamoorthi and Hanrahan 01b]. Illumination and reflectance reconstruction from photos and videos remains to be an active research area in computer graphics and computer vision.

Realistic Rendering In order to achieve consistent overall appearance when compositing virtual objects into real-world footage, not only direct scene illumination must be known but also inter-object light transport between all objects in the scene has to be computed. Inter-object light transport gives rise not only to cast shadows but also to such visually important yet subtle effects as indirect illumination, color bleeding, and caustics. In essence, taking these effects into account amounts to computing the rendering equation for the entire augmented 3D scene, Chapter 6.

Although stunningly realistic images can be rendered interactively (e.g., in video games), the light transport in these scenes is often approximated or based on simplifying assumptions [Ritschel et al. 12]. Rendering photorealistic images offline (i.e. without tight computational budgets but with significantly higher quality demands) is nowadays almost exclusively done using (Markov Chain) Monte Carlo methods. These methods share the concept of stochastically constructing paths that connect the sensor of a virtual camera to a virtual light source and computing the energy reaching the sensor's pixels. This process can be done in many different ways: sampling only from the sensor or the light sources (path tracing [Kajiya 86] or light tracing [Arvo 86]), sampling from both sides with deterministic connections (bidirectional path tracing [Lafortune and

Willems 93, Veach and Guibas 94]), mutating paths with Metropolis light transport [Veach and Guibas 97], or density estimation of path vertices (photon mapping [Jensen 96]). Going beyond pure light transport, additional realism can be achieved by simulating the effects of actual cameras such as depth-of-field [Lee et al. 10b, Kán 12], lens flares [Hullin et al. 11], or accurate simulation of lens models [Hanika and Dachsbacher 14].

Global illumination has seen tremendous progress in the last decades [Pharr and Humphreys 10, Křivánek et al. 13, Dachsbacher et al. 13]. Nevertheless, not all techniques are equally well suited for all scene settings which can require specifically tailored solutions [Veach and Guibas 97, Jakob and Marschner 12, Kaplanyan et al. 14, Hachisuka et al. 08, Kaplanyan and Dachsbacher 13a, Kaplanyan and Dachsbacher 13b, Georgiev et al. 12, Hachisuka et al. 12]. The demands on the rendering methods increase with the richness of detail, accuracy, and the spectrum of materials.

Compositing The most successful algorithm to cope with the insertion of virtual objects into a given scene is presumably the differential rendering technique first proposed in [Fournier et al. 93] and made popular by [Debevec 98]. The idea behind differential rendering is to compute the light interaction between the scene and the virtual object, i.e. the near-field illumination (for example, shadows cast on the ground), by making use of a coarse (hand-made) representation of the scene surrounding the virtual object. The scene is rendered once with and once without the virtual object to be augmented, and the difference is applied to the original input image. Given an input image I_{bg}, the (potentially manually created) 3D scene geometry proxy is global illumination-rendered from the same viewpoint as I_{bg}, once without the virtual object and once with the object inserted, resulting in images I_{noobj} and I_{obj}, respectively. Additionally, an object matte α is computed to mark pixels depicting the virtual object as 1 and all remaining pixels as 0. The final composite is then computed via

$$I_{\text{final}} = \alpha \cdot I_{\text{obj}} + (\mathbf{1} - \alpha) \cdot (I_{\text{bg}} + (I_{\text{obj}} - I_{\text{noobj}})) \; .$$

In this form, it becomes clear that for each object the pixel value is simply copied from the rendered image I_{obj}. For the remaining pixels, if the virtual object does not affect its surrounding, I_{obj} and I_{noobj} are equal and the result is equal to I_{bg}. If I_{obj} is darker than I_{noobj} light is subtracted from the input photograph, introducing shadowed regions. On the other hand, if I_{obj} is brighter than I_{noobj} intensity is added signifying, for example, caustics. Several improvements of this technique have been proposed, e.g., for moving objects [Drettakis et al. 97], using final gathering [Loscos et al. 99], making use of differential photon mapping for refractions [Grosch 05] or to take near-field illumination into account [Grosch et al. 07].

17.5 Summary

Image- and video-based rendering can be categorized into purely image-based approaches, which directly synthesize new images by re-sampling and interpolating the captured data, and geometry-assisted approaches that exploit reconstructed depth information or higher-level models of the scene to guide the image synthesis process. Beyond pure view interpolation of static scenes, approaches for dynamic scenes and objects allow synthesizing new images in a space-time continuum. The categorization into purely image-based and geometry-assisted approaches is not to be understood as a fixed classification but rather aims at giving an overview on existing methods. Many approaches do not fit exactly into one single category but can be located somewhere in between. Combining image- and video-based rendering with traditional 3D rendering is an active field of research. The augmentation of real world-acquired scenes with virtually created content frequently requires specifically tailored methods and solutions.