EUROGRAPHICS 2019/ O. Bimber and A. Fusiello *Poster*

Voxelizing Light-Field Recordings

D. C. Schedl, I. Kurmi, and O. Bimber

Johannes Kepler University Linz, Austria

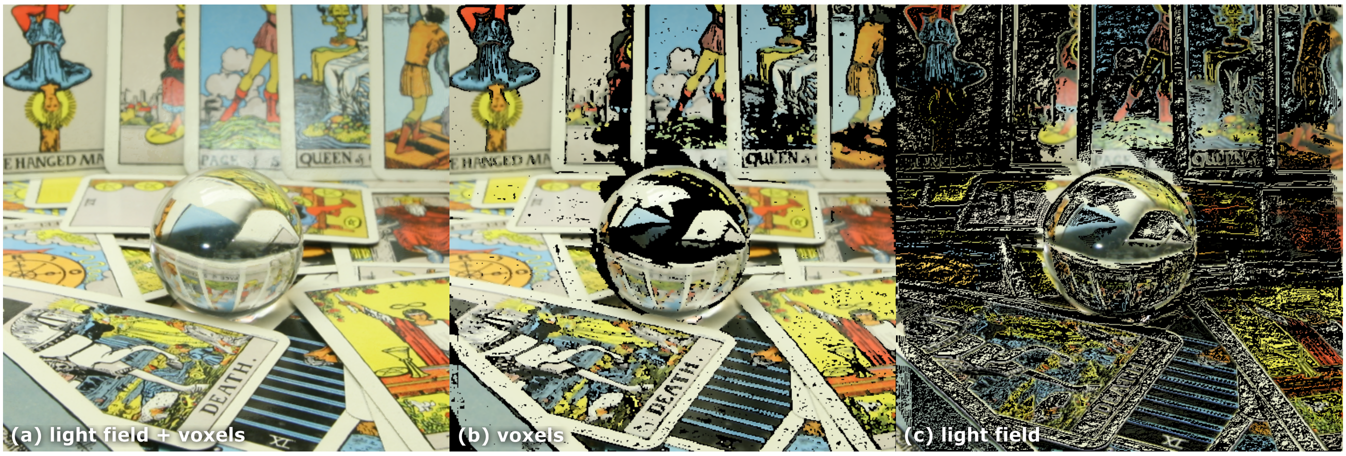

Figure 1: *Hybrid rendering of light-field and voxelized light-field data (a). The original light field consumes* 216.8 MB *of memory (*3 *colors;* 8 bit *per channel), while our hybrid representation compresses the data to* 82.4 MB *(*38 %*). Isotropic and uniform scene areas can be represented as voxels (color+depth), requiring only* 0.5 MB *of sparse data, without a perceivable loss of quality (b). Anisotropic scene parts (e.g. reflections and refractions) and occlusion boundaries (producing errors in depth reconstruction) are represented as sparse light field, requiring* 81.9 MB *of data (c).*

Abstract

Light fields are an emerging image-based technique that support free viewpoint navigation of recorded scenes as demanded in several recent applications (e.g., Virtual Reality). Pure image-based representations, however quickly become inefficient, as a large number of images are required to be captured, stored, and processed. Geometric scene representations require less storage and are more efficient to render. Geometry reconstruction, however, is unreliable and might fail for complex scene parts. Furthermore, view-dependent effects that are preserved with light fields are lost in pure geometry-based techniques. Therefore, we propose a hybrid representation and rendering scheme for recorded dense light fields: we extract isotropic scene regions and represent them by voxels, while the remaining areas are represented as sparse light field. In comparison to dense light fields, storage demands are reduced while visual quality is sustained.

CCS Concepts

• ***Computing methodologies*** → ***Computational photography; Image-based rendering;*** *Image compression;*

Due to the recent resurgence of Virtual Reality (VR), there is a demand for immersive experiences, where users can freely change their viewpoint (e.g., for Head-Mounted Displays). Typically, game engines or 360°images provide content for VR applications. While real-time renderings of game assets lack the visual fidelity of recorded scenes, monoscopic or stereoscopic 360° recordings lack the necessary motion parallax and do not store any view-dependent effects (e.g., the change of reflections and refractions as the viewer moves). Advanced image-based representations, like light fields, are suitable for capturing and rendering such complex effects. Pure image-based representations, however quickly become inefficient, as for large scenes an unrealistic number of images are required to be captured, stored, and processed. Especially mobile devices, that might be used in VR setups, lack the computational power to

DOI: 10.2312/egp.20191040

handle such amounts of data. In comparison to light fields, a geometrical representation requires less storage and is more efficient to render (cf. Figure 1). Reconstructing geometry based on captured images (e.g., structure from motion, multi-view stereo, etc.), however, is only reliably feasible for diffuse and textured surfaces. It might fail for transparent, reflecting, or uniform scene parts. Furthermore, color information retrieved from images do not contain view-dependent effects [ESG99]. Thus, we believe that instead of relying only on pure geometric representations which fail for complex scene regions, or only on an extensive light-field representation that requires excessive image sampling for large scenes, a hybrid representation and rendering scheme is most efficient. While the majority of common scenes can be represented with geometry, complex regions are covered with a light-field representation (cf. Figure 1).

Current hybrid rendering and reconstruction techniques support the computation of high quality images and free viewpoint navigation by combining 3D scene representations and high-resolution imagery. In comparison to conventional geometry-based rendering, fine scene details and view-dependent effects are preserved in recorded images. In contrast to classical small-baseline image-based approaches, such as classical light-field rendering, fewer images are required. This results in reduced artefacts at occlusion boundaries and preserved image discontinuities. Early work includes surface light fields [CBCG02] and surface reflectance fields [VSG*13], which add simple geometric proxies (such as spherical proxies) for light-field or reflectance-field parameterization to an underlying triangle mesh. Since these approaches require many image samples, there is no data reduction. More recent approaches [OCDD15,HRDB16,KKSM17] address these limitations by depth-based view-interpolation which supports larger baselines, but do not support anisotropic effects in cases where depth-reconstruction fails. Nevertheless, all existing hybrid rendering and reconstruction techniques still rely on an underlying geometric representation. If geometry reconstruction fails or the depth is not preliminary available (e.g., as for cinematic renderings), all previous hybrid rendering techniques will fail. In contrast to previous approaches, our goal is to segment the captured light field of a scene into regions that can be represented by geometry, and regions that cannot. Regions where geometry reconstructions fails will be represented and rendered as a light field that is seamlessly blended with the rendered geometry. Furthermore, our aim is to reduce the amount of data that needs to be stored and processed.

Our algorithm takes a densely sampled light field as input and extracts isotropic scene regions that can be represented by color and depth into a voxel representation, while the remaining areas are represented as sparse light field. We start by rendering the light field with different focal depths from front to back and store the result in a focal stack. Additionally, we compute a directional consistency value (i.e., the color variance over all directions) for every point on every slice of the focal stack, as described in other plane-sweeping algorithms (e.g., [ZC04]). The directional variance will be high in out-of-focus areas, at occlusion boundaries, and at anisotropic regions. For every pixel we determine the layer of the focal stack where the corresponding (orthographic projected) entry has minimum directional variance. This is done to remove the contributions of out-of-focus areas. All of these entries are then used as voxels in our hybrid representation if the minimum directional variance is less than a predefined threshold. These are rendered into the pixels that are likely to be in focus and represent isotropic scene areas, as shown in Figure 1 (b). Next we remove the rays from the original light field that intersect at these voxels. The remaining light-field rays represent anisotropic scene areas and occlusion boundaries, as shown in Figure 1 (c). We render the sparse voxel and light-field representations with the same camera parameters and blend the results as illustrated in Figure 1 (a).

Figure 1 shows renderings of our hybrid technique for the Tarot[†] light field of size $512 \times 512 \times 17 \times 17$ rays. Our algorithm is implemented in Matlab, we used 50 slices for the focal stack, and a variance threshold of 7.7×10^{-4} for each color channel ranging from 0 to 1. Renderings from different viewpoints are shown in the supplementary video.

Our current light-field voxelization approach has several shortcomings: First, variance thresholding has to be set manually and is not robust against miscalibrations in the recorded light-field data. Perceptual metrics that allow a comparison of intermediate results with the original light field might enable for automatic thresholding. Second, we currently do not use optimized data structures for sparse data representations. While they exist for sparse volume data sets, we need to investigate efficient data structures for sparse light fields that support compressed storage as well as efficient reading and rendering. Furthermore, we need to develop real-time rendering techniques for consistent visualization of both scene components. In comparison to classical light-field rendering, we expect vast improvements in rendering speed, due to the decreased computational demand mainly caused by the reduced light-field size.

References

[CBCG02] Chen W.-C., Bouguet J.-Y., Chu M. H., Grzeszczuk R.: Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Trans. Graph. 21*, 3 (2002), 447–456. 2

[ESG99] Eisert P., Steinbach E., Girod B.: 3-d shape reconstruction from light fields using voxel back-projection. In *Proc. Vision, Modeling, and Visualization Workshop* (1999), pp. 67–74. 2

[HRDB16] Hedman P., Ritschel T., Drettakis G., Brostow G.: Scalable inside-out image-based rendering. *ACM Trans. Graph. 35*, 6 (2016), 231:1–231:11. 2

[KKSM17] Koniaris C., Kosek M., Sinclair D., Mitchell K.: Real-time rendering with compressed animated light fields. In *Graphics Interface* (2017). 2

[OCDD15] Ortiz-Cayon R., Djelouah A., Drettakis G.: A bayesian approach for selective image-based rendering using superpixels. In *International Conference on 3D Vision* (2015). 2

[VSG*13] Vanhoey K., Sauvage B., Genevaux O., Larue F., Dischler J.-M.: Robust fitting on poorly sampled data for surface light field rendering and image relighting. *Computer Graphics Forum 32*, 6 (2013), 101–112. 2

[ZC04] Zhang C., Chen T.: A self-reconfigurable camera array. In *ACM SIGGRAPH Sketches* (2004), p. 151. 2

[†] http://lightfield.stanford.edu/lfs.html