Light-Field Supported Fast Volume Rendering

Clemens Birklbauer[1] and Oliver Bimber[1]
Johannes Kepler University Linz

Abstract

Introduction and Motivation

Advances in imaging technology lead to ever larger image data sets. Modern scanning microscopes, for instance, produce image stacks with a megapixel lateral resolution and with possibly many hundreds to thousands of slices in axial direction. This trend will continue, resulting in very large volumetric data sets that are difficult to explore interactively because the complexity of volume rendering is proportional to the spatial and lateral resolution of the data.

Light-field rendering is a fast and simple image-based rendering method that requires precomputed or precaptured image data. For volume rendering, each expensively computed image is discarded after viewing parameters change, while the renderer is idle if the viewing parameters do not change and the visualization need not to be updated.

Light-Field Supported Fast Volume Rendering

We present a combination of light-field and volume rendering to enable high-quality interactive explorations of large volumetric data sets. We use the idle times of the volume renderer for filling a cache-managed light field. New images are then composed from high-resolution light-field rendering and from volume rendering – depending on the state of the light-field cache. Our method leads to better image quality than the low level of detail that can be achieved by a volume renderer at the same frame rate.

Light-Field Caching

Visualizing high resolution volumetric datasets in adequate quality requires billions of light-field rays to be computed, processed and stored in real-time with limited graphics memory. This is possible with our light-field caching framework described in [1]. The light-field data is split into pages of pixels blocks that are managed by the cache. Only the pages required for rendering an image are held in the graphics memory. A probability based caching strategy determines the likelihood of a page to be required for rendering based on interaction predictions and maintains the cache accordingly. To enable surround navigation we support a spherical light-field parameterization.

Related Work

Since light-field rendering is a low-cost image-based technique, it has been used in combination with volume rendering before. Approaches such as that presented in [2,3] convert large volumetric datasets into a light-field representation in an offline pre-processing step. The light-field data is then used for rendering the volumetric content at interactive frame rates that cannot be supported by online volume rendering. We integrated our light-field caching method into a volume renderer to support interactive viewing of large volumetric datasets. In contrast to previous approaches, we apply light-field caching, and use the volume renderer's idle times to fill and update the cache dynamically on the fly, based on interaction predictions made by our probability-based prefetching strategy.

Using volume caching methods (such as texture bricking with octree space partitioning), current GPU-implemented volume renderers achieve fast frame-rates and adequate image quality if the data required for rendering one frame fits entirely into the graphics memory. For large, but mainly opaque and sparse volumes real-time rates are possible, as shown in [4]. However, highly detailed and transparent volumes (as created by scanning microscopes, for example) are significantly larger and complex rendering methods (e.g., when simulating physically based light transport with Monte Carlo ray tracing) make real-time rates difficult to achieve for these cases. Since lateral and axial resolutions of scanning microscopes are increasing continuously, we believe that a dynamically updated light-field cache and integrated image-based rendering as part of a volume renderer can be beneficial to the interactive exploration of such data sets.

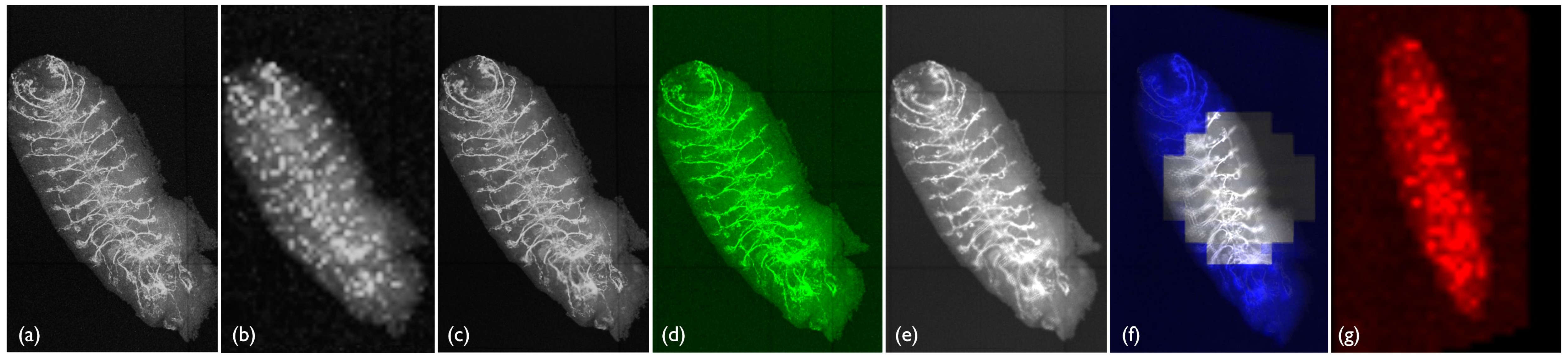

Figure 1: Visualization of a drosophila (4,096x4,096x61 volume resolution, 2.86 GB – 6,016x6,016x31x31 light-field resolution, 34.8 gigarays, 97.17 GB, rendered using a spherical parameterization): full-resolution volume rendering at 0.4 fps (a), volume rendering preview at 25 fps (b), our method at 25 fps (c). Color-coded contributions of different sources during rotation (d-g): full-resolution volume (green), volume preview (red), full-resolution light field (gray), and fallback light field (blue). The visible seams are the result of the microscope's scanning process, not of visualization.

Combined Rendering

Since in our case volume rendering is largely independent from light-field rendering, an arbitrary volume renderer can be used in combination with light-field caching. If the volume renderer already supports own acceleration strategies, this does not conflict with our approach but increases the overall rendering performance. Many volume renderers switch to a fast preview mode with a lower level of detail (LOD) to achieve interactive frame rates during user interactions. We, however use our cache-managed light-field rendering to support a much higher LOD at the same frame rate.

When the user stops interacting with the volume (i.e., at constant viewing parameters), the volume renderer first computes and displays a full-resolution image for the currently rendering camera. When otherwise idle, it renders pages of the cached light-field data structure in the background. For this purpose, we use virtual perspective cameras that are uniformly distributed on a bounding sphere U which encloses the volume (cf. figure 2a). These cameras point towards the volume center and have a field of view that includes the bounding box of the volume. Note that since pages cover only portions of the field of view of a perspective camera, they are rendered by constructing their individual sub-frusta (cf. figure 2b).

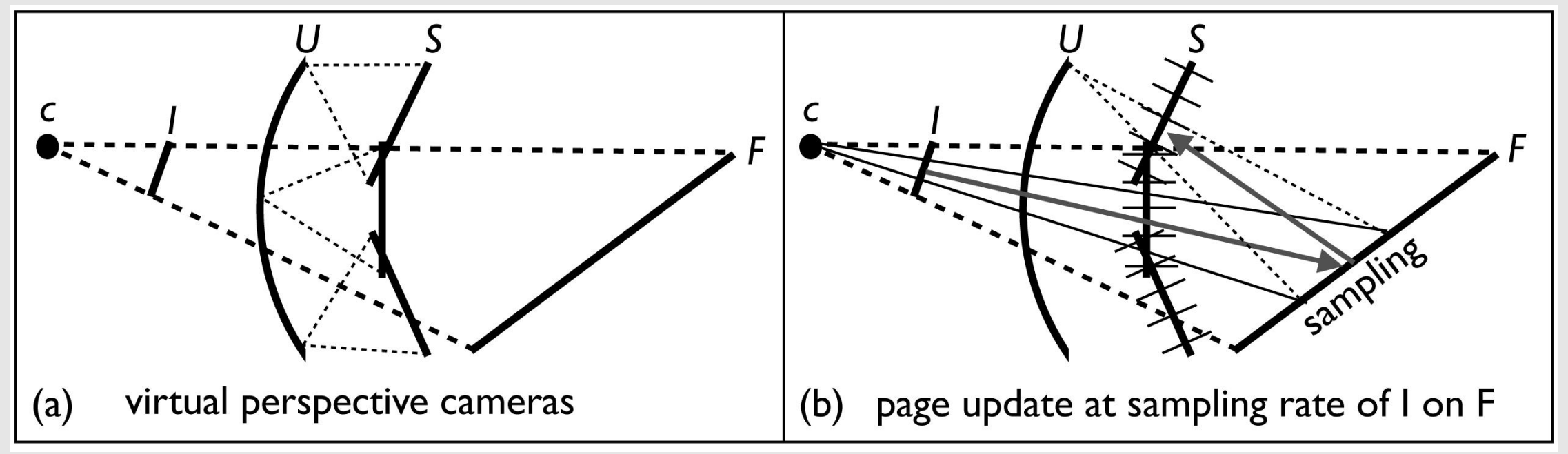

Figure 2: Spherical parameterization (a) and dynamic page update (b). The current position of the render camera is denoted by *c*, *I* is the image plane of the camera, *U* is the bounding sphere and *S* the image plane of a virtual perspective camera on *U*. The splitting of *S* into pages is illustrated in (b).

The sampling rate that we use for rendering each page (i.e., the ray casting resolution of the volume renderer) is chosen such that it matches the sampling rate of the rendering camera at a common reference plane inside the volume. We chose the focal plane F of the rendering camera for this (cf. figure 2b). Note, that this sampling rate equals our LOD. The page probabilities, as computed for the light-field caching, determine the proper order in which the pages are rendered into the cache. Pages with a high probability are rendered first. If a page already exists in a higher or equal LOD than required for the current rendering camera, it needs not to be re-rendered and remains valid. Only pages with a lower LOD, or non-existing pages are updated. After the necessary updates, the remaining idle time is used for a second update round (in priority order). This time, pages are updated with the highest possible LOD – which is the ray casting resolution that optimally samples the most distant voxels covered by the page.

For fast image rendering during user interaction (i.e., while changing the viewing parameters), we compute the per-pixel LODs that are achieved by light-field rendering (using the pages of the full-resolution light field and, if necessary, the pages of a fallback light field with a lower resolution that can be generated quicker) and by the fast preview of the volume renderer. For the latter case, the LOD is simply determined as described above. But since in light-field rendering, each pixel contains the contributions of multiple pages, we average the (potentially different) LOD values of all pages involved. The final image is then assembled from pixels of the sources that have the highest LOD per pixel (either the full-resolution / fallback light field or the volume rendering preview). Note that light-field caching and volume rendering preview are configured to achieve the same minimum target frame rate. Figure 1 illustrates an example.

Results

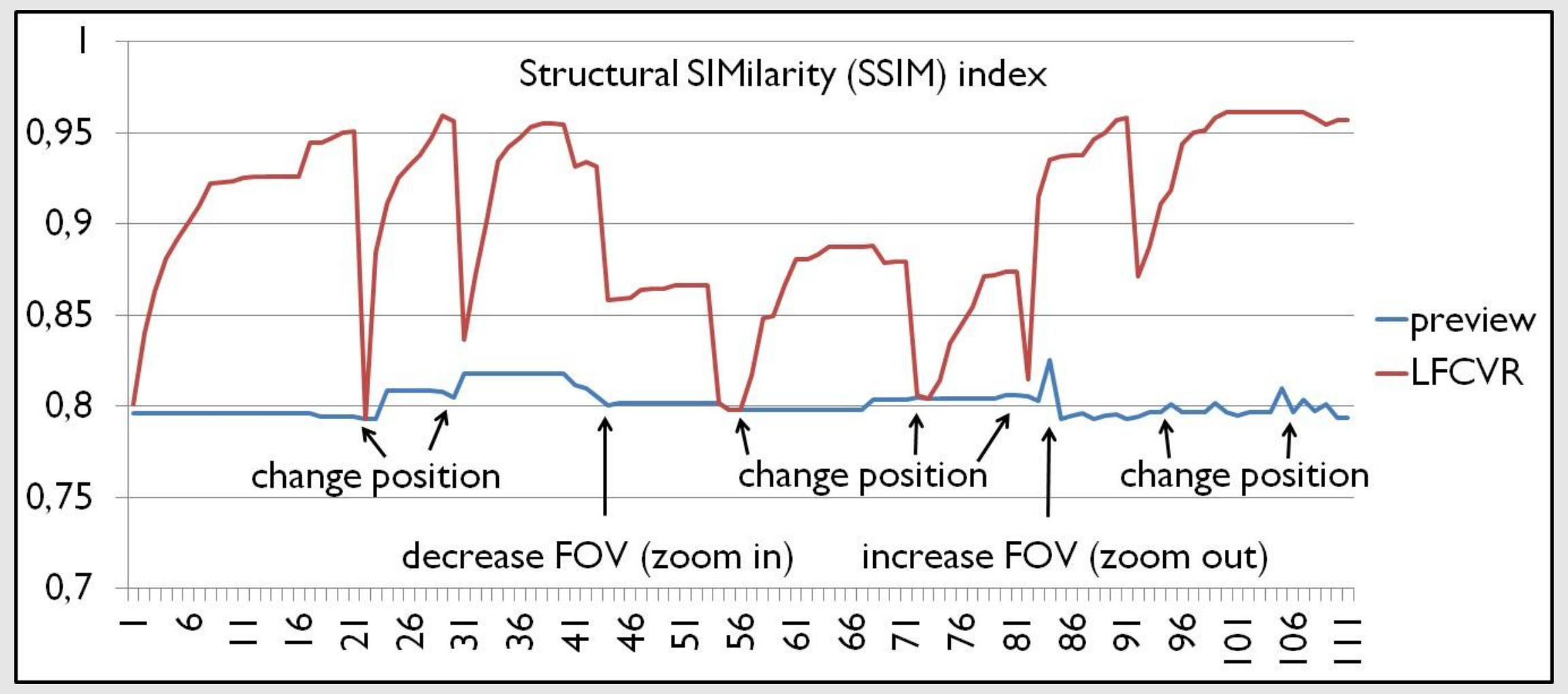

Figure 3: Measured SSIM index (y-axis) when comparing light-field-cached volume rendering (red) or volume rendering preview (blue) with the full-resolution volume rendering (ground truth) for a sequence of frames (x-axis: time in seconds) during an identical navigation trace at the same average frame rate of 25 fps. A SSIM index of 1.0 indicates a perfect match. For visual reference: the SSIM values when comparing figures 1b,c with figure 1a are 0.79 and 0.96, respectively.

Figure 3 illustrates for the z-stack shown in figure 1 that our light-field-cached volume rendering leads to better image quality than the low-level-of-detail preview of a volume renderer at the same frame rate (25 fps in this example, on a Quad-Core 2.67GHz with NVIDIA GeForce GTX 580). We measured the Structural SIMilarity (SSIM) index [5] (y-axis) between images rendered by our method (LFCVR), or by the volume renderer at 25 fps (preview), and the corresponding full-resolution volume rendering at 0.4 fps for a sequence of frames (x-axis: time in seconds) during identical navigation traces (using a spherical parameterization). For volume rendering, we used the Visualization Toolkit (VTK). Any faster volume renderer will lead to a quicker filling of the light-field cache and therefore to a faster transition from volume rendering to light-field rendering. Light-field-cached volume rendering only becomes inefficient if the volume renderer approaches the quality and performance of light-field rendering.

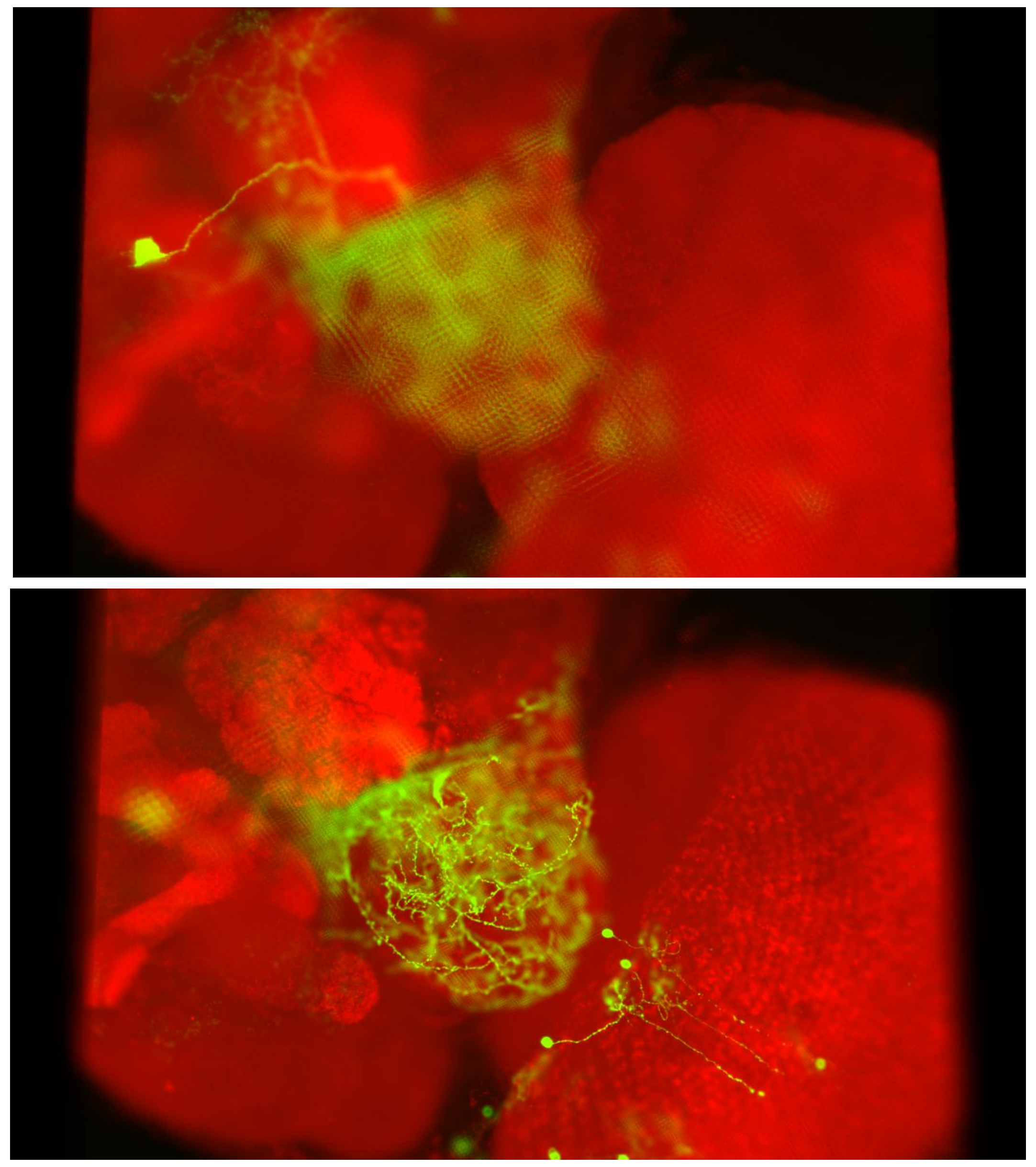

Figure 4: Refocusing allows highlighting features at different depths and is an alternative to conventional slicing. It supports focus and context visualization.

Limitations and Future Work

Uniform and insufficient sampling of light-field perspectives (i.e., samples on U) leads to sampling artifacts (i.e., ghosting) in the rendered images. On-demand creation and dynamic management of new and unstructured light-field perspectives based on the user interaction can reduce these artifacts. A faster volume renderer (e.g., running on a second GPU) can lead to an additional speed-up and a quicker transition from volume rendering to light-field rendering.

However, light-field-cached volume rendering is useful only for navigation tasks. If other rendering parameters (such as the transfer function) are changed, then the light-field cache must be reset and newly filled. Thus, for frequent changes in rendering settings other than the viewing parameters, light-field-cached volume rendering will not be more efficient than standard volume rendering.

References

1. Opelt, S. and Bimber, O.: Light-Field Caching, ACM Siggraph (poster), 2011
2. Vlasic, D., Pfister, H., Molinov, S., Grzeszczuk, R., and Matusik, W.: Opacity light fields: interactive rendering of surface light fields with view-dependent opacity. In Proceedings of the 2003 symposium on Interactive 3D graphics, ACM (2003), 65–74.
3. Rezk-Salama, C., Todt, S., and Kolb, A.: Raycasting of light field galleries from volumetric data. Computer Graphics Forum 27, 3, (2008), 839–846.
4. Crassin, C., Neyret, F., Lefebvre, S., and Eisemann, E.: Gigavoxels: ray-guided streaming for efficient and detailed voxel rendering. In Proceedings of the 2009 symposium on Interactive 3D graphics and games, ACM (2009) 15–22.
5. Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13, 4,(2004), 600–612.

---

[1] {firstname.lastname}@jku.at

More information at: www.jku.at/cg