Light Field Caching
Simon Opelt¹ and Oliver Bimber²
Johannes Kepler University Linz

Introduction

With the continuously increasing sensor resolutions of cameras, light field imaging is becoming a more and more practical extension to conventional digital photography. It complements postprocessing by synthetic aperture control, refocussing, as well as perspective and field of view changes. For being a true alternative to classical 2D imaging, however, the spatial resolution of light fields must be in the same megapixel order as the resolution of today’s digital images. The additional angular resolution must also be adequately high to prevent sampling artifacts (in particular for synthetic re-focusing). This will quickly cause gigapixels rather than megapixels of data that have to be rendered with limited graphics memory. We describe a light-field-caching framework that makes it possible to render very large light fields in real-time.

Related Work

Caching and prefetching can be applied for rendering high-resolution 2D images. The approach described in [1], for example, manages image tiles with a client-side cache during interactive rendering.Tiles neighboring the current view on the 2D image plane are prefetched after the required data has been loaded. Caching in the context of light fields has been used for efficient light field decompression. The uncompresed blocks of an MPEG-encoded light field are cached during rendering to avoid repetitive decompression of image regions. An index table is used to manage blocks of data. The hit ratio of the cache is reduced during sudden changes of the view.

Because many light-field representations and parameterizations use collections of 2D images another possibility for caching is to use a per-image basis as described in [3]. Based on an estimated viewing frustum a set of images that are likely required next can be determined. A large cache unit size, however, is inefficient for situations where small regions of data in many different images are required for sampling. Light-field caching can also be applied as an alternative (image-based) approach to volume rendering. In [4] a large volumetric dataset is converted to an opacity light-field representation during an offline preprocessing step. The data is then used for rendering initially complex datasets at interactive frame rates.

Our Approach

We have developed a CUDA-based light-field renderer that is extended by a software-managed cache. In a preprocessing step the light-field data is split into pages of 128 x 128 pixels (configurable) which results in 4MB blocks that are managed by the cache and that are stored in the main memory on hard disk. Similar to virtual memory and virtual texturing our system maintains an indirection structure (i.e., an index table) for each block of data that has to be loaded. Caching the required pages is done by sampling a new image from the light field. In particular, it is a projection from the perspective camera which makes up the light field over the adjusted focal plane into the aperture of the virtual camera.

For determining which pages should be loaded into the cache and which pages should be evicted if the cache is full different strategies can be applied. We have implemented a CUDA-accelerated version of Least Recently Used (LRU) as a reference caching strategy. For each frame the required pages are computed and compared with the state of the cache for deciding which data is additionally required. If no free cache slots are available, LRU replaces the pages that have been least recently used.

In addition to a simple LRU strategy, we investigated a probabilistic approach that determines the likelihood of pages to be required in future frames as an alternative decision metric for loading, evicting, and additional prefetching. It applies classical re-ranking to a set of previously adjusted virtual camera parameters (position, aperture, focus, and field of view) to estimate the future camera parameters. For each of the estimated future parameters we determine all pages that are required for all values within an (adjustable) range around the estimated parameter space. The further the distance of these values from the corresponding estimation the lower are their probabilities. This leads to a set of parameter-individual probabilities that are assigned to the associated pages. The overall probability of a page is the weighted sum of all parameter probabilities. The weighting allows for amplifying and attenuating the importance of individual parameters. The pages are then loaded in the order of their overall probabilities, starting with the pages that are required in the current frame. Thus, pages with higher probabilities are prefetched earlier than pages with lower probabilities.

The amount of pages that can be transferred per frame is limited for a given frame-rate. A rapid change in viewing the light field (e.g., caused by fast camera movements, or aperture/focal-length ensembles) may require large amounts of data to be loaded at once. As soon as missing pages cannot be transferred fast enough, this leads either to a dropping frame rate or to a visual degradation of the rendered image that is sampled from the incomplete light field. We therefore keep track of the number of pages that are missing per frame and use this as an error metric for evaluation.

We believe our approach to be applicable for any light field format as well as other view-dependent applications.

Results

Sample results are presented in figure 2. The light field shown has a size of 4.3 GB and is rendered with 30fps using an allocated 450MB cache on an NVIDIA GeForce GTX 480. An on-demand streaming of required data without caching leads constantly to missing data for rendering (or slower frame rates). With the same angular resolution, the spatial resolution would have to be downscaled to 512 x 512 to fill all the data into graphics memory and to render the light field appropriately without caching. We are currently investigating optimal weighting strategies and additional compression options for our probability-based caching approach. Our implementation currently also enables stochastic rendering of very large light fields in real-time.

Applications

Besides the rendering of high-resolution light field data that has been captured with imaging devices (e.g., high resolution camera arrays or future compact light-field cameras) our method also supports the visualization of very large z-stacks that are acquired with scanning microscopes (cf. figure 4).

References


© 2011 Johannes Kepler University Linz

More information at: www.jku.at/cg