Remote Rendering for Ultrascale Data

UltraVis
SciDAC Institute for Ultrascale Visualization

Sandia National Laboratories

Kenneth Moreland **Daniel Lepage** **David Koller** **Greg Humphreys**

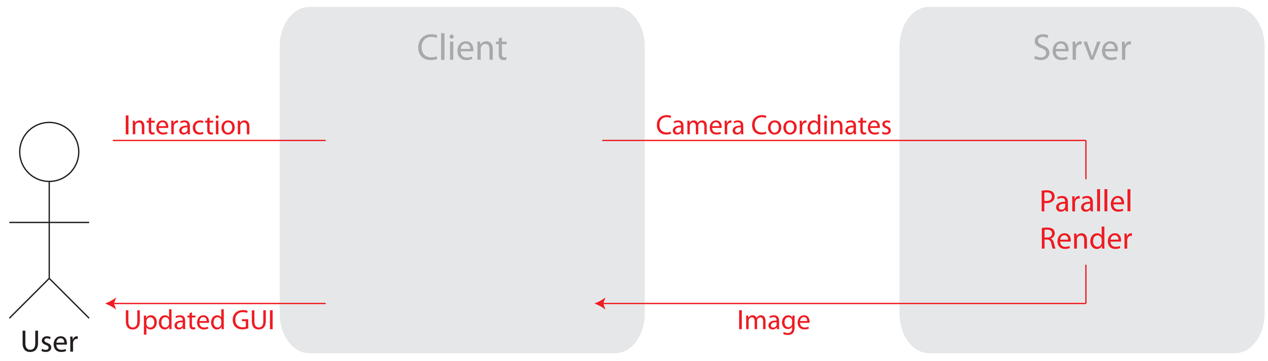
Sandia National Laboratories University of Virginia

Abstract

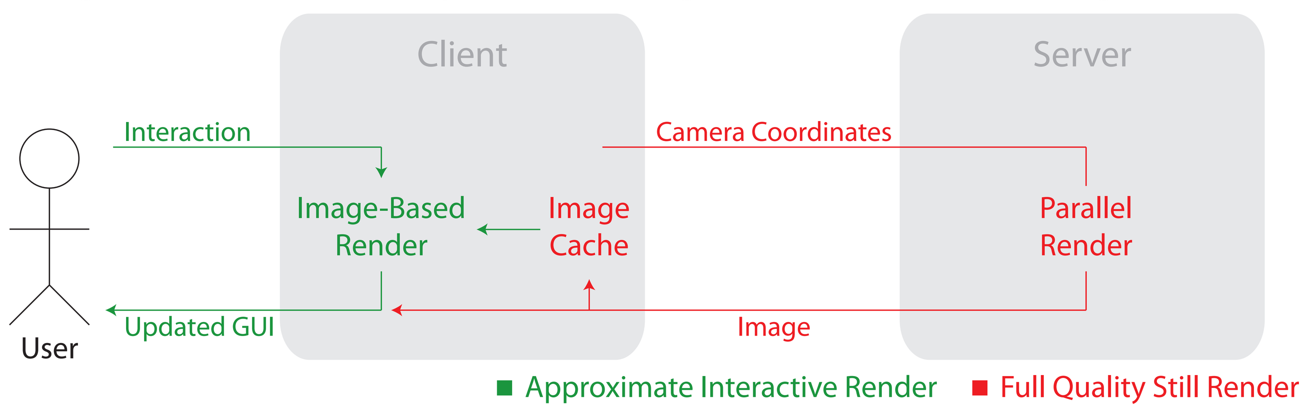
The mission of the SciDAC Institute for Ultrascale Visualization is to address the upcoming petascale visualization challenges. As we move to petascale computation, we are seeing a trend not only in the growth but also in the consolidation of computing resources. As the distances between user and petascale visualization resources grow, the expected performance of the network degrades, especially with respect to latency. In this paper we will explore a technique for remote visualization that leverages unstructured lumigraph rendering. This technique will provide an interactive rendering experience regardless of the network performance to the remote visualization resource. The unstructured lumigraph rendering can also replace many of the other level-of-detail techniques currently used that have problems that are exasperated by petascale data.

Remote Rendering

Current parallel rendering techniques and many other parallel visualization techniques scale well enough to handle petascale data [Ma, Wylie]. The new focus on visualization is in loading, managing, and processing data. One consequence of this is the practice of co-locating a visualization resource with the supercomputer for faster access to the simulation output [Ahern, Cedilnik]. Co-locating the visualization resource with the supercomputer necessitates the ability to perform remote visualization, which in turn necessitates in an interactive loop for rendering.



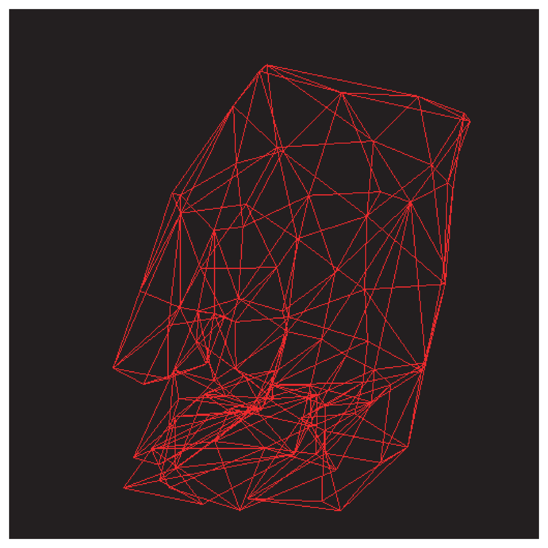
When the client and server are connected over a wide area network, the latency of the network may make interactive rendering from the server impossible. We are investigating a new technique for approximate interactive rendering in remote visualization using image based techniques. High-quality still images generated by the server are cached on the client. This cache of images is used to synthesize images from new viewpoints during interactive rendering.



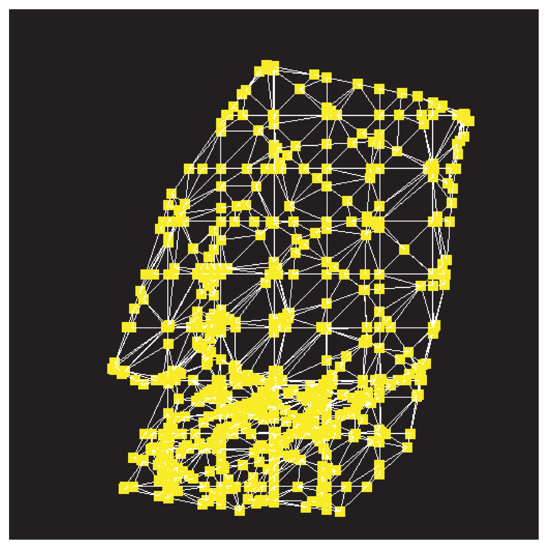
Using images rendered from the server to synthesize other approximate images has two distinct advantages. First, it is scalable with respect to the size of the geometry being represented. Like the sort-last parallel rendering employed on the server, the overhead of the technique is independent of geometry size. Second, the image base rendering can leverage the rendering already being done on the server. Lengthy preprocessing of the data is not necessary, as is the case with geometric decimation.

Unstructured Lumigraph Rendering

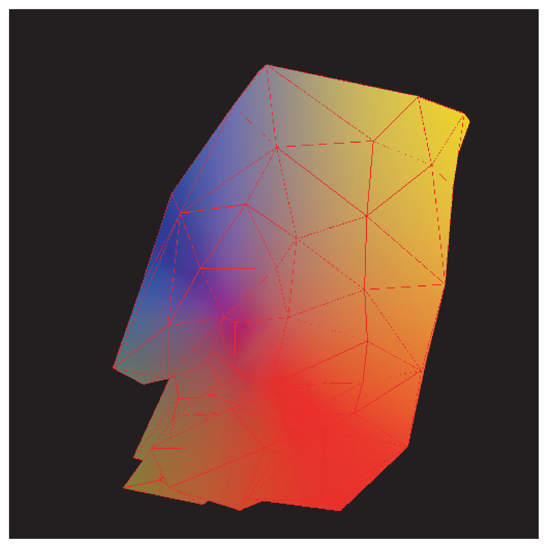
We perform the client-side image-based rendering by extending the Unstructured Lumigraph Rendering technique [Buehler]. Images are generated by projecting the input images onto a geometric proxy and merging the resulting images according to the blending weights.



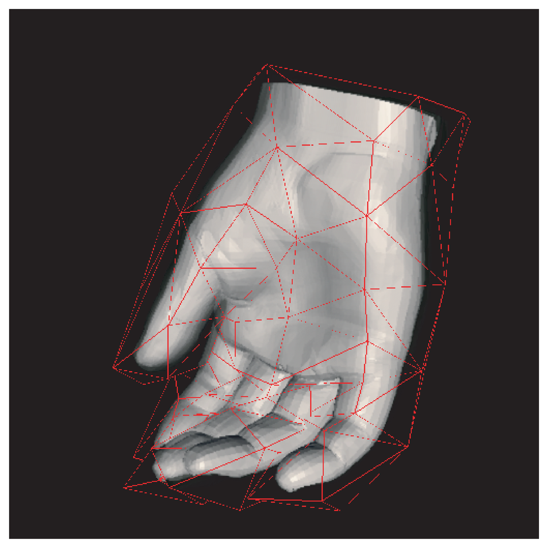
We use a 3D triangle mesh proxy (for example at left) to map pixels from prior rendered images to pixels in the image plane. The quality of the synthesized image increases when the proxy closely approximates the shape of the actual dataset; however, the overhead increases as well.



Rather than recomputing the blending field at every point in the image plane, we sample at every vertex of the visible proxy faces, at the center of projection for every known view, and in a regular grid across the image plane (shown at right, top). We then perform a Delaunay triangulation in the plane constrained to edges of the proxy and regular grid.



To project the samples back to the proxy, we trace viewing rays through the samples to the face of the containing proxy. From there we can interpolate the blending field on the proxy (shown at right, bottom).

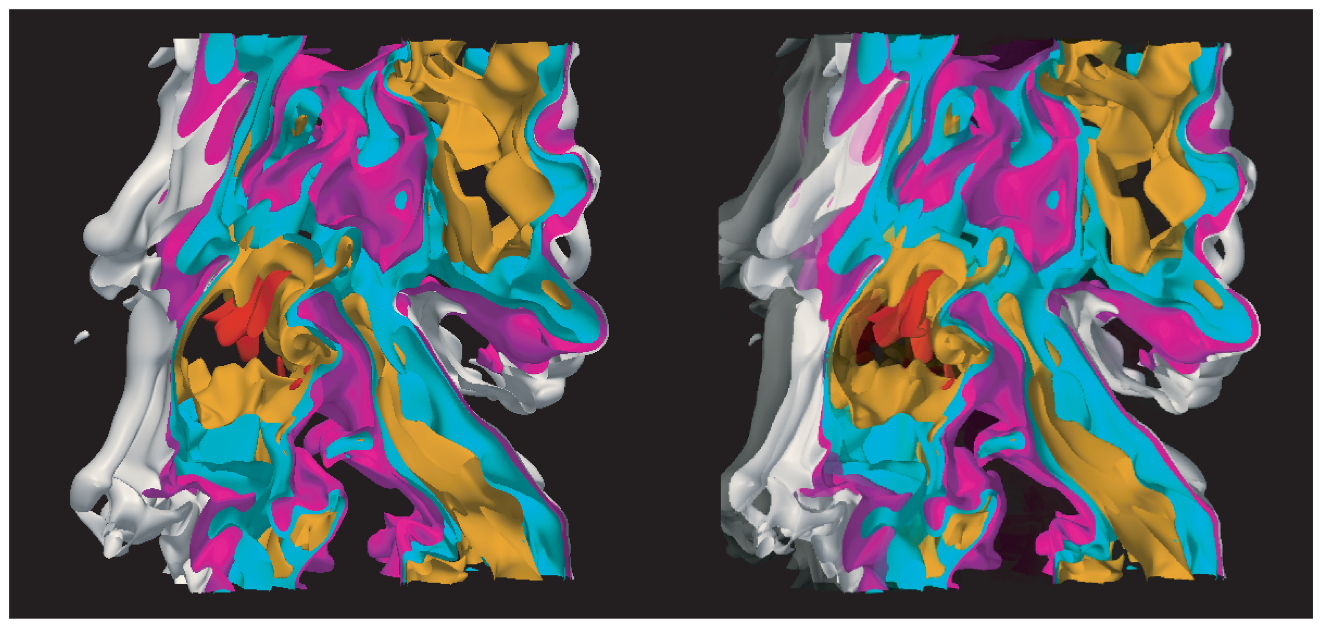


We assign each input image a weight at each sample point based on the similarity between the current view and the view used to generate the image. Each sample is rendered once for each image with a non-zero weight using texture mapping and the renderings are blended together (as shown at left).

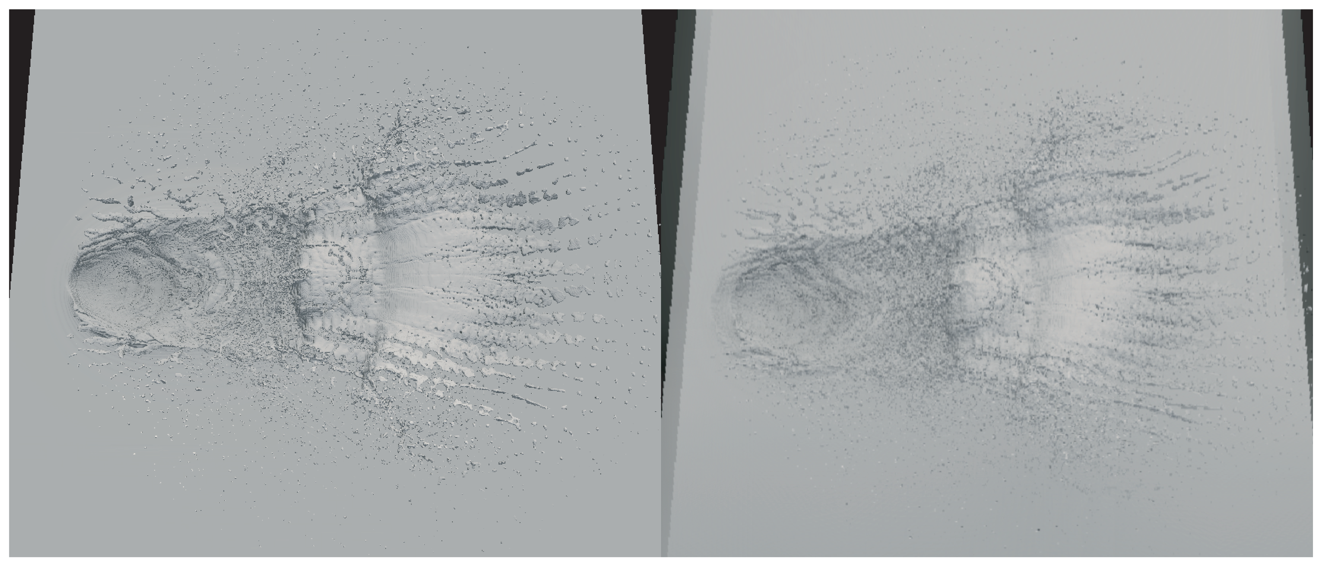
Lumigraph techniques improve when input images near the desired view are available. We are developing "prefetching" techniques to predict the user's navigation path through the 3D environment and during breaks in the navigation render images from viewpoints most likely to be visited in the near future.

Application to SciDAC Projects

Our current work involves applying the image based rendering techniques to large scale scientific data, which is frequently encountered within the SciDAC projects. Much of this work involves integrating lumigraph rendering into the ParaView application, which is already capable of the remote visualization of large data sets. Below is an example of using ParaView for viewing a 41 million cell time-dependent simulation of combustion turbulence (data courtesy of Jacqueline Chen, Sandia National Laboratories). The left image shows the full resolution rendering; the right image shows a lumigraph approximation that is rendered locally.



As another example, consider how well the lumigraph approximates this complicated surface of material fragmentation with no knowledge of the underlying geometry.



Our current challenges include adding multithreading to the rendering routines to allow us to "prefetch" images while ParaView sits idle waiting for user. We also need robust and accurate ways to generate proxy geometries of general data sets.

Acknowledgments

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

Ahern S 2007 Petascale visual data analysis in a production computing environment *Journal of Physics: Conference Series (Proceedings of SciDAC 2007)* vol 78

Buehler C, Bosse M, McMillan L, Gortler S and Cohen M 2001 Unstructured lumigraph rendering *Computer Graphics (Proceedings of SIGGRAPH 2001)* 425–432

Cedilnik A, Geveci B, Moreland K, Ahrens J and Farve J 2006 Remote large data visualization in the ParaView framework *Eurographics Parallel Graphics and Visualization 2006* pp 163–170

Ma K L 1994 Parallel volume rendering using binary-swap image composition *IEEE Computer Graphics and Applications* 14 59–68

Wylie B, Pavlakos C, Lewis V and Moreland K 2001 Scalable rendering on PC clusters *IEEE Computer Graphics and Applications* 21 62–70