Bridging the Resolution Gap: Superimposition of Multiple Multi-Channel Volumes

Chih K. Liang, Jürgen P. Schulze
Calit2, UCSD

Ruth West, Maryann Martone
NCMIR, UCSD

Matthias Zwicker
CSE, UCSD

We present a new algorithm to render multiple overlapping volume data sets in real-time. Our approach uses textured slices for rendering volumes, and GLSL shaders for the application of transfer functions and the blending of data channels.

The National Center for Microscopy and Imaging Research (NCMIR) images biological systems at multiple scales and resolutions using a variety of techniques. Visualization across scales is useful to scientists because 1) they often must combine multiple microscopies with different resolutions to understand a specimen; 2) it is more efficient to take lower resolution views of broad expanses of tissue and higher resolution views of features of interest. Combining these together allows detail to be appreciated in context. Generating data sets at different levels of spatial resolution does not provide the precise coordinates required to co-locate, or superimpose the resulting volumes automatically. Our algorithm allows users to interactively co-locate multiple nested volumes in real-time. Figures 1 to 3 demonstrate our approach using volumes from ultra-wide field light microscopy (LM) and high resolution electron microscopy (EM).

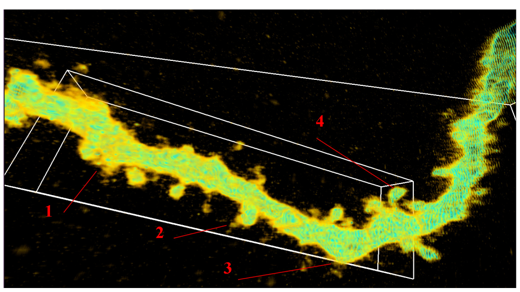

Figure 1: LM: $512 \times 246 \times 128$ voxels

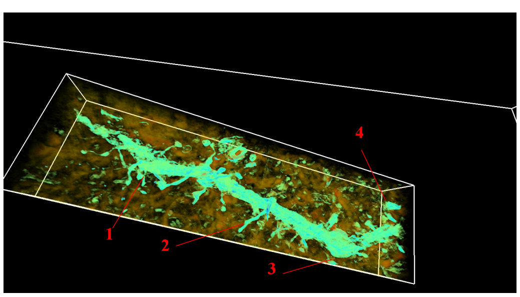

Figure 2: EM: $256 \times 512 \times 128$ voxels

Slice interleaving. For every volume, 2D textured slices parallel to the screen are calculated from the volume in graphics hardware, utilizing the approaches for 3D texture-based and multi-channel volume rendering, as described in [Cullip and Neumann 1993] and [Schulze and Rice 2004]. Our algorithm interleaves the textured slices of all volumes and renders them from farthest to closest with respect to the viewer. Whenever slices of different volumes are rendered we switch to the respective transfer function and shader of the current volume.

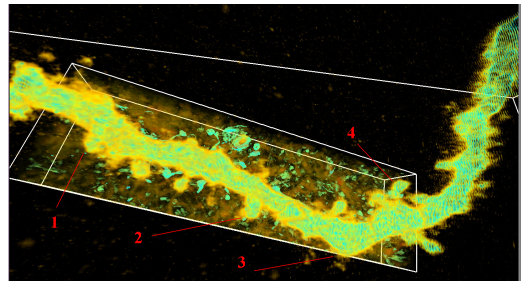

Figure 3: LM and EM superimposed

Blending channels within voxels. Every voxel contributes RGBA values to the frame buffer when rendering textured slices. The color values (RGB) are determined directly by the channel values, i.e., channel #1 determines the intensity of red. At this point our algorithm supports up to three channels. The maximum value of all channels determines the voxel's opacity. A GLSL shader is applied based on the number of data channels in the volume.

Blending voxels from different volumes. Our algorithm blends textured slices onto the frame buffer using either alpha blending, or maximum intensity projection (MIP). We use the standard blending options OpenGL provides in this process.

Performance results. We measured the frame rates for each volume in Figures 1 to 3 including their combination and list the results in Table 1. For this experiment we used a dual AMD Opteron PC with an Nvidia Quadro 4500 graphics board.

Volume	# Slices	Frame rate
LM	208	30 fps
EM	312	51 fps
LM+EM	520	19 fps

Table 1: Performance results for LM, EM, and LM+EM. We rendered in a window of 512×512 pixels.

Future Work. In our experience, aligning multiple 3D volumes with conventional 2D displays is quite difficult due to occlusion and a lack of positional cues. We plan to overcome these limitations by implementing the software in an immersive, tracked virtual reality environment.

References

CULLIP, T., AND NEUMANN, U. 1993. *Accelerating Volume Reconstruction With 3D Texture Hardware*. Technical Report TR93-027, University of North Carolina, Chapel Hill.

SCHULZE, J., AND RICE, A. 2004. *Real-Time Volume Rendering of Four-Channel Data Sets*. In poster proceedings of IEEE Visualization 2004, Austin, TX.