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

Chih K. Liang, Jürgen P. Schulze
California Institute for Telecommunications and Information Technology
University of California San Diego
{cliang, jschulze}@ucsd.edu

Ruth West, Maryann Martone
National Center for Microscopy and Imaging Research
University of California San Diego
{rwest, maryann}@ncmir.ucsd.edu

Matthias Zwicker
Department of Computer Science of Engineering
University of California San Diego
matthias@graphics.ucsd.edu

Overview.

- Goal: To render multiple multi-channel volume data sets in real-time.
- Utilizes textured slices for rendering volumes and GLSL pixel shader for the application of transfer functions and the blending of data channels.

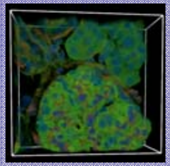
Motivation.

- Multiple scales and resolutions.
- Combine multiple microscopies with different resolutions to understand a specimen.
- More efficient to take lower resolution views of broad expanses of tissue and higher resolution views of features of interest.
- Allows detail to be appreciated in context.

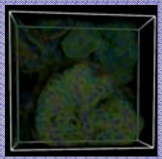
Interaction.

- 2 options for blending textured slices: Maximum Intensity Projection (MIP), Alpha Blending.
- 3 types of transfer function for mapping channel data or opacity values:

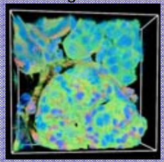
Gamma Function



High-Pass Filter



Histogram CDF



- Different color schemes – Every data channel can be represented by shades of a color.

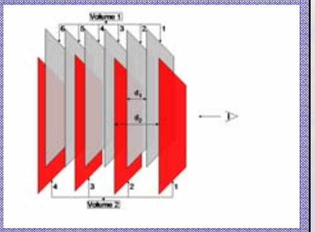
  Most popular is Red, Green, Blue for 3-channel volumes (as in pictures above); the right image shows Cyan, Magenta, Yellow as channel colors.

- Multiple volumes can be manipulated by translation and rotation individually.

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 volume rendering [1]. The algorithm interleaves the textured slices of all volumes and renders them from farthest to closest with respect to the viewer [2]. Whenever slices of different volumes are rendered, it switches to the respective transfer functions of the current volume

```
for all slices s[k] do
    for each volume v[i] do
        Compute farthest slice v[i].fs
        if v[i].fs.depth > s[k].depth then
            s[k] := v[i].fs
        endif
    endfor
    if s[k].volume != s[k-1].volume then
        Switch transfer functions and etc
    endif
    Render s[k]
endfor
```



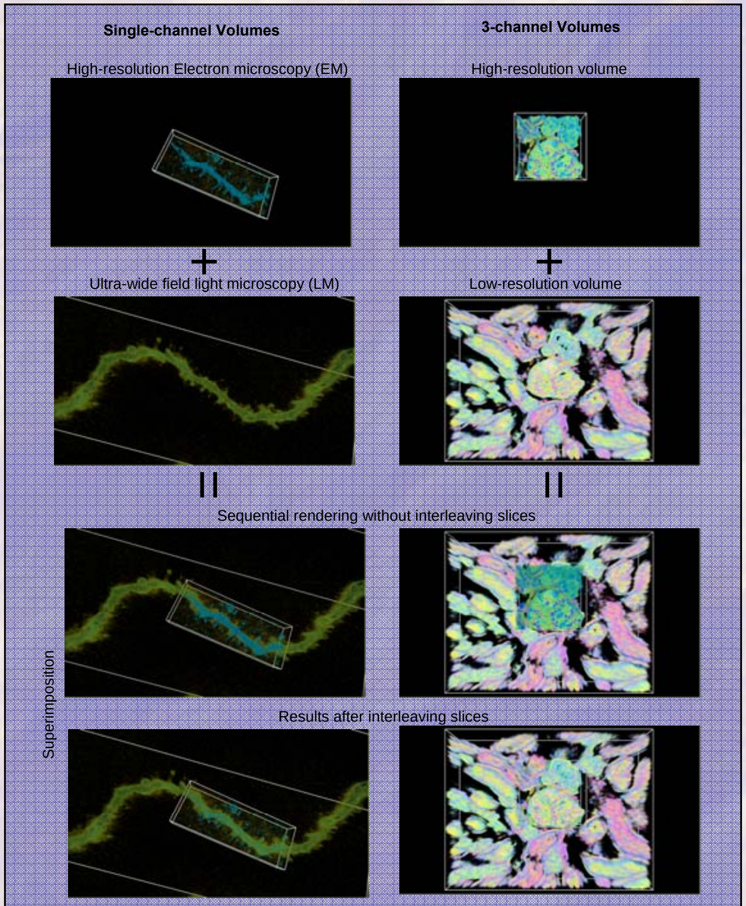
Blending channels

- The pixel shader uses textures as look-up tables.
- The data channels are separated into different 3D texture maps.
- A volume has one transfer function for every channel and only one opacity transfer function.
- The shader outputs color and opacity values for a voxel with multiple data channels.

$$C^{shader} = \sum_{j=1}^{n} I \times W_j \times f^C(D_j) \times C_j$$

$$\alpha^{shader} = f^o\left(C^{shader} \bullet \left\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right\rangle\right)$$

n = number of channels in the volume
I = intensity specified by the user for the volume
W_j = weight specified by user for j^{th} channel
f^C = transfer function for the data channel
D_j = data value of the j^{th} data channel
C_j = color specified by user for j^{th} channel
f^O = transfer function for the opacity



Results.

- Application tool implemented with OpenGL, GLUT and GLUI gui interface.
- Machine specifications: Intel Core 2 duo @3.15GHz, 2GB of RAM, 320MB 8800GTS video card, Windows XP 32-bit.
- Performance results (rendered in 1600x900 windows):

Single-channel Volumes

	LM	EM	LM+EM Sequantial	LM+EM Interleaving
# Voxels	512x246x128	256x512x128	N/A	
Voxel Size	0.165x0.12x0.15	0.03x0.04x0.02		
# Slices	128	128	256	
Render Time	23.634 ms	4.911 ms	26.936 ms	27.016 ms
— GL Draw	0.195 ms	0.049 ms	0.355 ms	0.399 ms
Non-Render Time	0.349 ms	0.441 ms	0.706 ms	0.864 ms
— Slice Sorting	No Sorting			0.049 ms
Frame Rate	~42 fps	~188 fps	~36 fps	~36 fps

3-channel Volumes

	High-Resolution	Low-Resolution	Superimposition Sequantial	Superimposition Interleaving
# Voxels	256x256x128	512x256x128	N/A	
Voxel Size	2.0x1.875x1.39	2.9x4.5x1.34		
# Slices	128	128	256	
Render Time	10.221 ms	86.057 ms	96.947 ms	94.177 ms
— GL Draw	0.047 ms	0.047 ms	0.096 ms	0.670 ms
Non-Render Time	0.338 ms	0.275 ms	1.013 ms	1.429 ms
— Slice Sorting	No Sorting			0.052 ms
Frame Rate	~95 fps	~12 fps	~10 fps	~10 fps

Future Work.

- 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.
- Implementation of multi-resolution algorithms will increase the size of a volume that can be displayed, which is currently limited by the GPU memory size.

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

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

[3] RSLER, F., TEJADA, E., FANGMEIER, T., ERTL, T., AND KNAUFF, M. 2006. *Gpu-based multi-volume rendering for the visualization of functional brain images.* In T. Schulze, G. Horton, B. Preim, and S. Schlechtweg, editors, SimVis, pages 305–318. SCS Publishing House e.V.

SIGGRAPH2007