

Supplemental Material for SIGGRAPH 2004 Poster
fMRI Visualization of Multiple Functional Areas

Jan Hardenbergh, TeraRecon, Inc.

jch@terarecon.com

This document contains all of the text and most of the pictures which will be used for the 60x40 inch poster to be displayed at SIGGRAPH 2004. However, the images are of lower resolution. The actual 12Kx8K poster image is available, as well as any follow on work. Check <http://www.jch.com/volumes/fmrviz.htm>

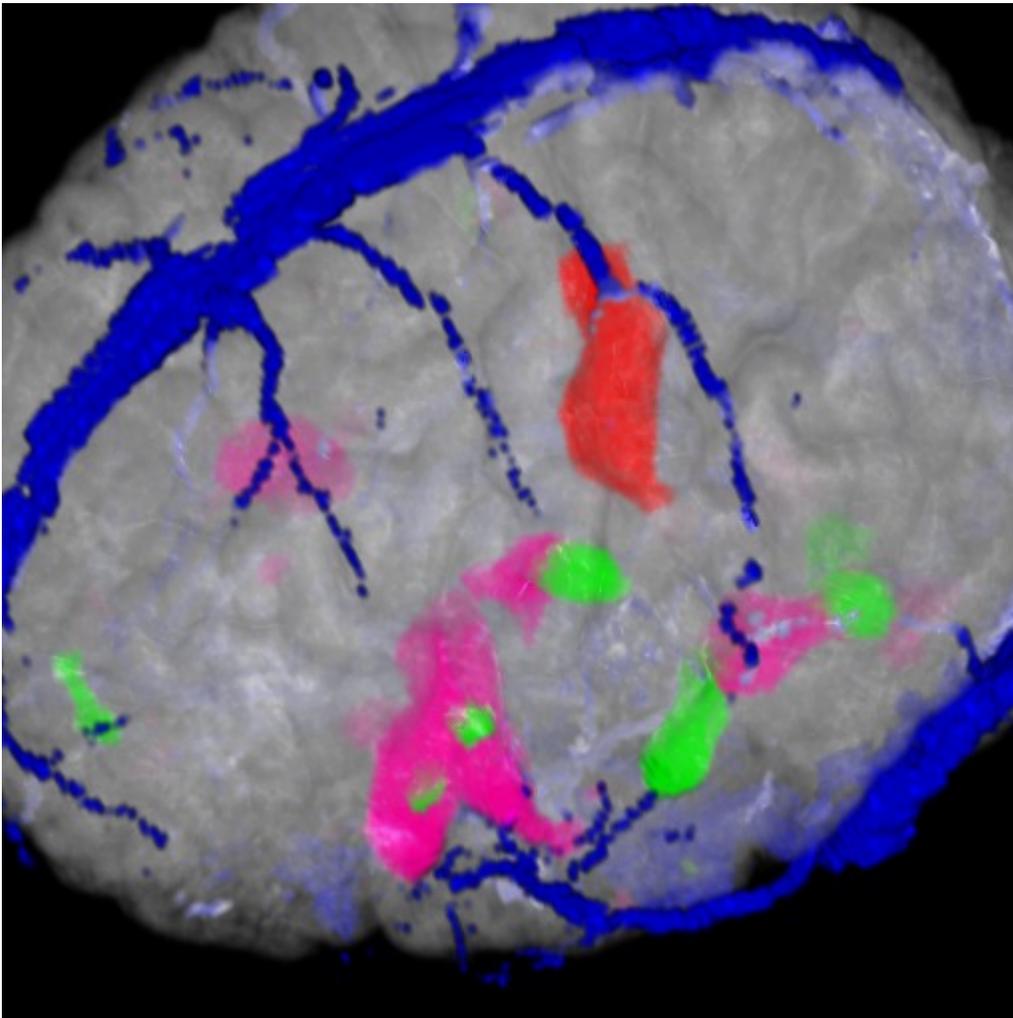


Figure 1. fMRI imaging of the brain allows us to see what areas of the brain are active during certain mental operations. In this image we see the anatomy, the vasculature in blue and three activation areas, one motor (red) and two language areas (green and magenta).

This poster presents a mixture of application needs, volume rendering, medical imaging, techniques and graphics hardware to visualize multiple functional areas in fMRI.

Background

Functional Magnetic Resonance Imaging (fMRI) displays different regions of the brain activated for different mental functions. Some of the identified functional areas are: language generation and acquisition, left and right motor control, touch sensation, visual processing, Broca's area as well as many more specific functions.

fMRI takes advantage of the change in blood flow to regions of the brain when they are active. With increased flow of fully oxygenated blood, there is a local reduction in deoxyhemoglobin, the "oxygen unloaded" form of hemoglobin. Since deoxyhemoglobin is paramagnetic, its absence can be identified in the strong magnetic field used for MR imaging.

The patient is given a series of mental exercises to perform while in the MRI scanner. These are designed to show the different functional areas of the brain. For example, tapping ones fingers will activate the motor control region for the left or right side.

We need to visualize the activation areas along with the brain anatomy and vasculature (blood vessels) to provide context for the activation areas. By turning on and off the different image components, rotating and clipping the volume, we can find optimal access paths to the tumor that avoid specific functional areas of the brain. This is extremely useful in neurosurgical planning.

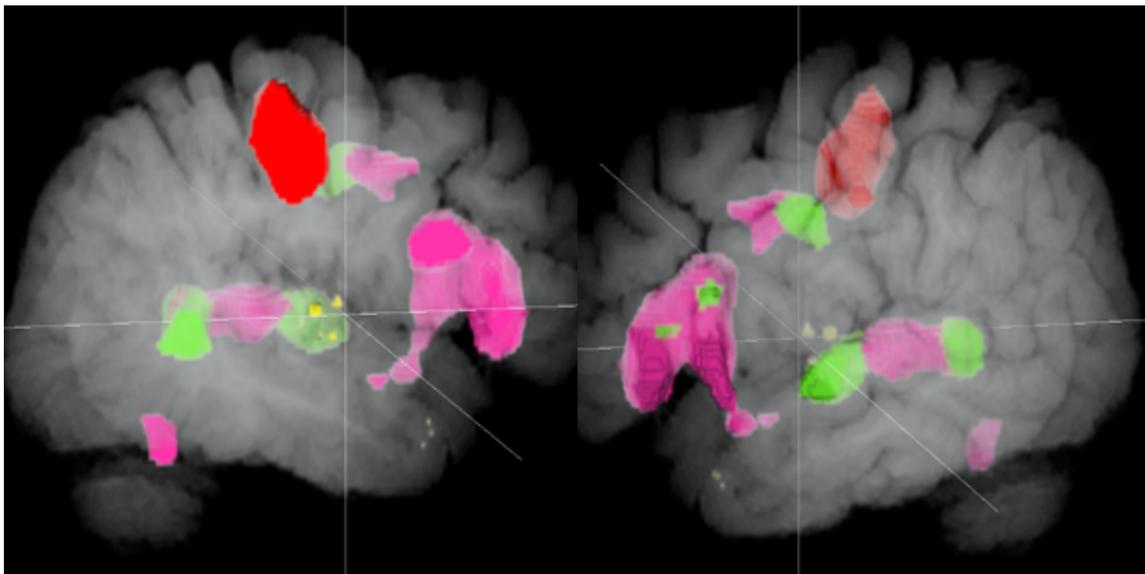


Figure 2: A large tumor is in the area at the center of the cross hair cursor. It contains the three yellow areas. On the left, we see the tumor from the inside of the brain. On the right, we see the tumor from the outside. The surgeon must avoid the functional areas in order to prevent noticeable impairments in the patient.

Activation Datasets

The reconstructed scan data consists of a volume dataset for each activation area as well as the anatomy and vasculature. We display the anatomy and vasculature to provide context for the activation areas. The example dataset has three activations: right hand motor, language acquisition, verbal generation. In all, we have five volumes. Each is 256x256x124 voxels; each voxel is 8 bits. That's 8MB each for a total of 40MB.

While it is possible to render all of these volumes together to create each image, it requires many texture reads and many pixel operations. For the anatomy and vasculature we have four reads and for each activation we need 3 texture lookups. To render an image with 3 activations requires a total of 13 texture reads followed by three serial pixel subtractions and 4 additions. This is not possible on the VP1000 [Wu 2003]. Also, memory requirements, complexity and performance on current graphics processors leads us to look for alternative solutions to displaying multiple activation areas.

Packing Multiple Activations

First, we create a volume dataset with three channels: anatomy, vasculature and activation areas. To create a single channel for activations, we combine datasets by adding an offset to each activation's nonzero integer voxel values. The offset is the sum of ranges of the previously processed activations. The dynamic range of each area is small, usually less than ten, so, with eight bits we should be able to display about 25 activations. This gives us a total data size of 24 MB regardless of the number of activations.

Some functional areas overlap, for example language generation and acquisition. Overlaps are handled by creating new activation areas. These are displayed as blends, if both areas are visible, or in the same color as the sole visible activation area.

In our example, we create a fourth activation, -VG-AC, which represents the overlap of the VG and AC activations. This is hidden by the application, although it can be treated the same as the other overlaps for debugging. The overlap area is displayed as a blend when both activations are visible, or in the color of the sole visible activation.

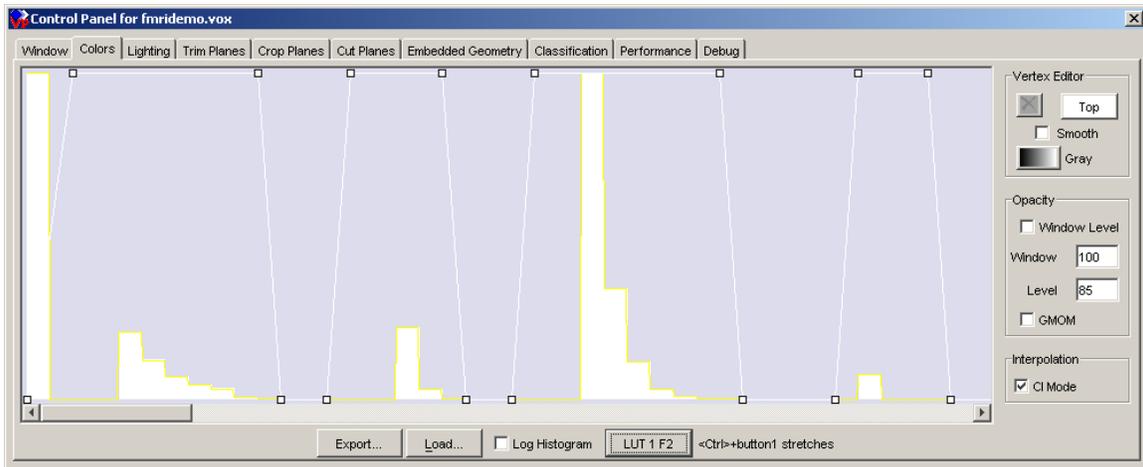


Figure 3: Histogram of the combined activation areas. The tall bar on the left is zero. The graph is normalized to the nonzero voxel values of the highest incidence - the first entry of VG (magenta). It is value 4 in its own dataset and value 24 in the combined dataset. The right most column is the fourth activation area created from the overlapping voxels from AC and VG. Below is a text description.

```
RHM: [4] 2067, [5] 1182, [6] 693, [7] 439, [8] 317, [9] 61, [10] 3
AC: [4] 2801, [5] 478, [6] 14
VG: [4] 10570, [5] 3586, [6] 1214, [7] 288, [8] 80, [9] 23, [10] 6
```

```
ALL: [4] 2067, [5] 1182, [6] 693, [7] 439, [8] 317, [9] 61, [10] 3, [11] 0, [12] 0, [13]
0, [14] 0, [15] 0, [16] 2216, [17] 301, [18] 8, [19] 0, [20] 0, [21] 0, [22] 0, [23] 0,
[24] 10049, [25] 3391, [26] 1162, [27] 288, [28] 80, [29] 23, [30] 6, [31] 0, [32] 0,
[33] 0, [34] 0, [35] 0, [36] 768
```

```
fmRI: fmrid.vox from SPGR Vascular VEINS RHM 2 4 9 AC 14 16 5 VG 22 24 9 -VG-AC 36 36 4
```

The histograms for RHM, AC, and VG show the original datasets. The ALL histogram is for the resulting packed Activation Area channel. The RHM values are unchanged; AC has been shifted from 4-6 to 16-18; VG from 4-10 to 24-30. Four values serve as a buffer between activation areas. The 768 voxels that are both in AC and VG have been transplanted to value 36. This results in some loss of gradation.

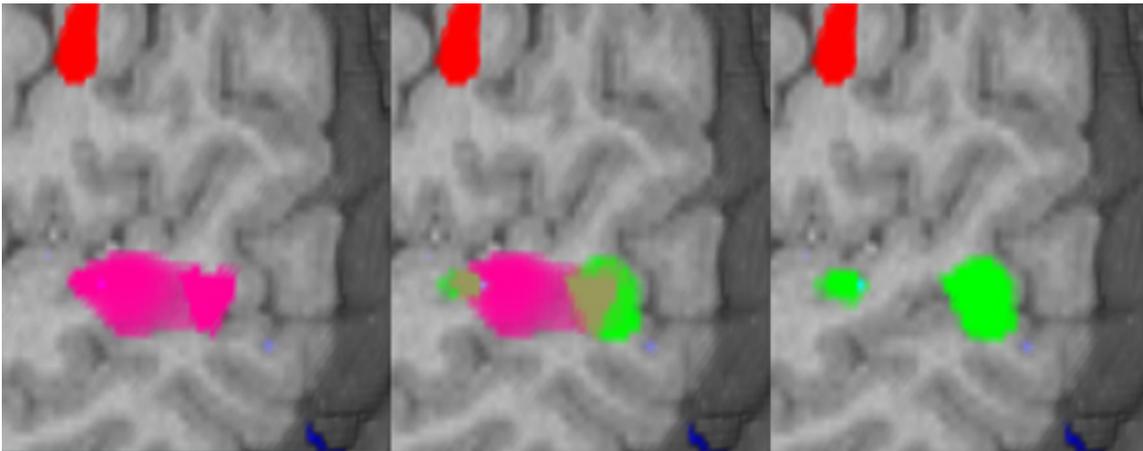


Figure 4: Overlap Display, notice in the middle image the merged color where the magenta and the green overlap. When only AC is visible, the overlap appears green, when it's VG, magenta.

Channel Masking

Four different "channels" can be rendered together efficiently on some hardware platforms. That allows us to see two activation areas in context. However, without proper masking the colors will bleed from one channel to the other. This bleeding is desirable between activation areas as it shows overlaps. However, when the activation areas and the anatomy blend, it becomes difficult to identify the activation areas.

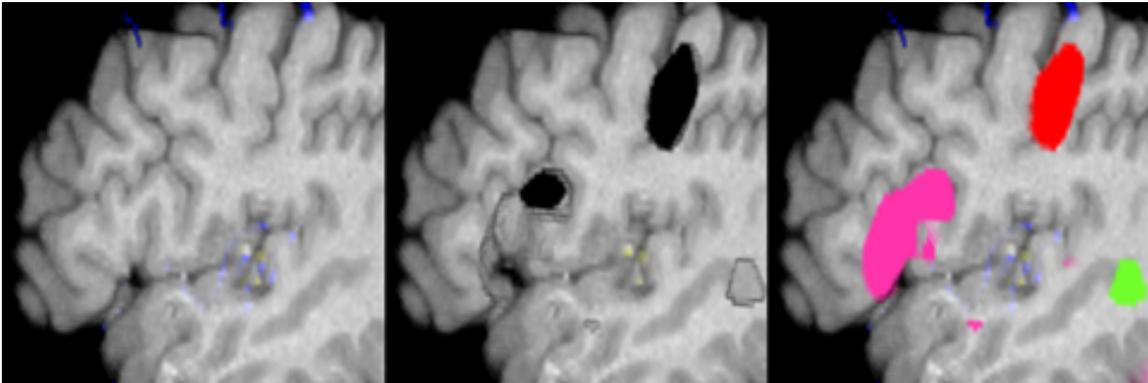


Figure 5: On the left we have the anatomy with some vascular data bleeding into it. In the middle is the anatomy with the activation areas subtracted. On the right is the final image.

To mask the anatomy channel with the activation areas, we create a mask LUT. This LUT has opaque white voxels for the values we are interested in. For example, to mask the RHM activation area in our example, we would fill the mask LUT from 2 to 11. When the opaque white is subtracted from the anatomy color, the resulting (clamped) color will be transparent black.

By using a single channel twice, once to mask the anatomy's color and once to add in the activation's color, we completely replace colors on a voxel by voxel basis and prevent blending between those two channels. It is tolerable that the vasculature bleeds into the anatomy when both are visible.

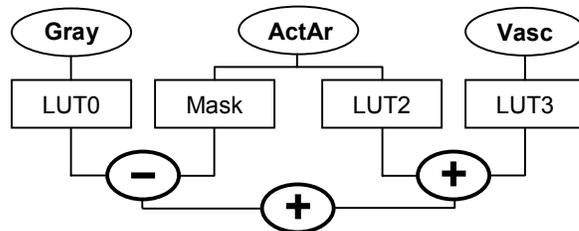


Figure 6. Channel Masking: the RGBA result of the anatomy is masked by the RGBA result of the Mask lookup. The color for the activation areas comes from a second lookup (LUT2). The VP1000 has 4 LUTs and 3 combiners that implement this operation.

Channel Masking on the ATI Radeon works in the same fashion. We used 1D textures for the LUTs:

- Read Gray (anatomy), Activation Areas, and Vascular channels in the first pass.
- Setup dependent read of 1D textures (LUTs)
- Read LUTs for Gray, Mask, Activation Area Color, and Vasculature.
- Subtract Mask from all channels of Gray/LUT result
- Add result + Act/LUT + Vasc/LUT

Interpolate First – IC vs. CI

When we are volume rendering the step of looking up the “voxel” value in a LUT is known as classification, as in we are classifying this voxel as a given tissue type. We are also interpolating because the rays that follow the pixels do not land on voxel centers. If we interpolate before we classify (IC) rather than classify and then interpolate (CI), we get a nicer image. This is the subject of controversy in the volume rendering world and ultimately resolves to signal processing.

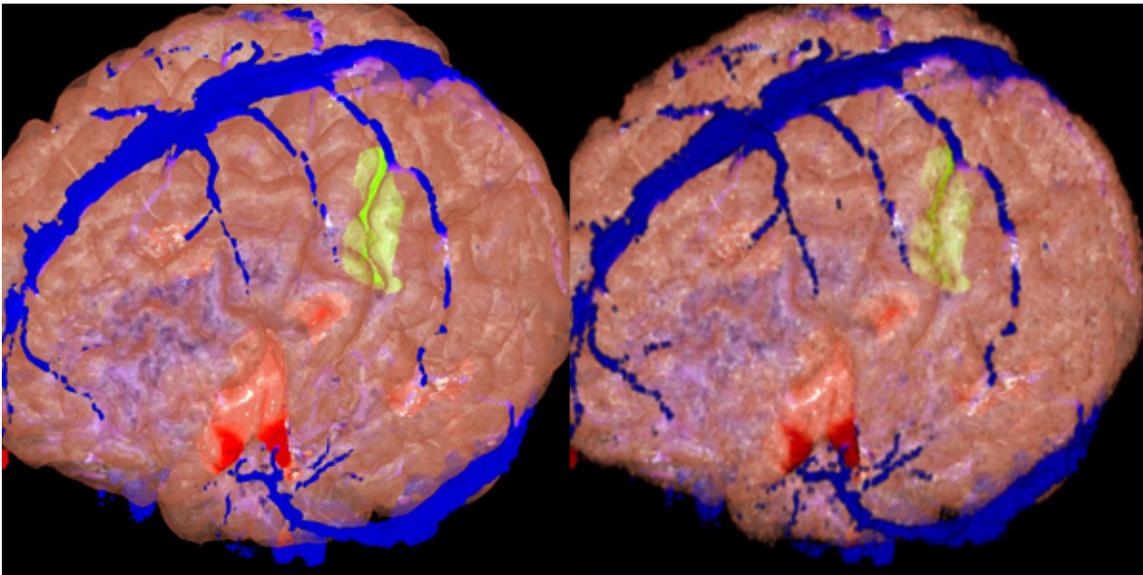


Figure 7: The image on the left is using IC mode and on the right is CI mode. Notice the surface of the cortex above the green functional area. When we interpolate the data first allows us to find the implicit surfaces embedded in the voxel mesh. Since this example is using one channel per component, it works in IC.

When we are actually rendering, we generate samples along the ray we traverse for each pixel. The sample is interpolated to be at a precise position. To get the value for that sample, we interpolate between the surrounding voxels. The samples and not the voxels are actually used for rendering.

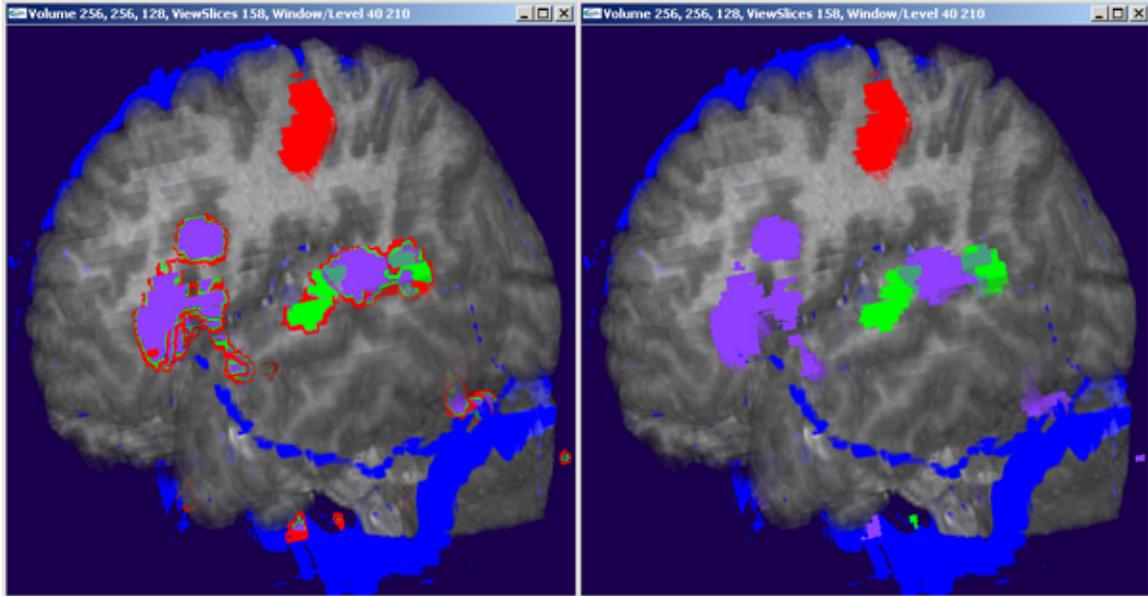


Figure 8: IC mode on the Radeon (left) and nearest neighbor (right).

When we pack the activation areas into a single channel, we invalidate interpolation of the voxel values. This is because when we interpolate from no activation to the AC activation, we generate samples that are displayed as the RHM and VG activations. This means we must use CI mode. Figure 8 shows examples from the Radeon. With IC we interpolate through other activation areas. This is really distracting and unusable. We can do a hack with nearest neighbor, which shifts the problem, but it's still very distracting. For "real" CI, we need to create RGBA on Radeon/NVIDIA, however, to do so doubles the amount of data. It was not possible to load the sample dataset with the activations as RGBA on the 64MB Radeon configuration I was using. On the VP1000, both IC and CI modes are built in and selectable without any preprocessing.

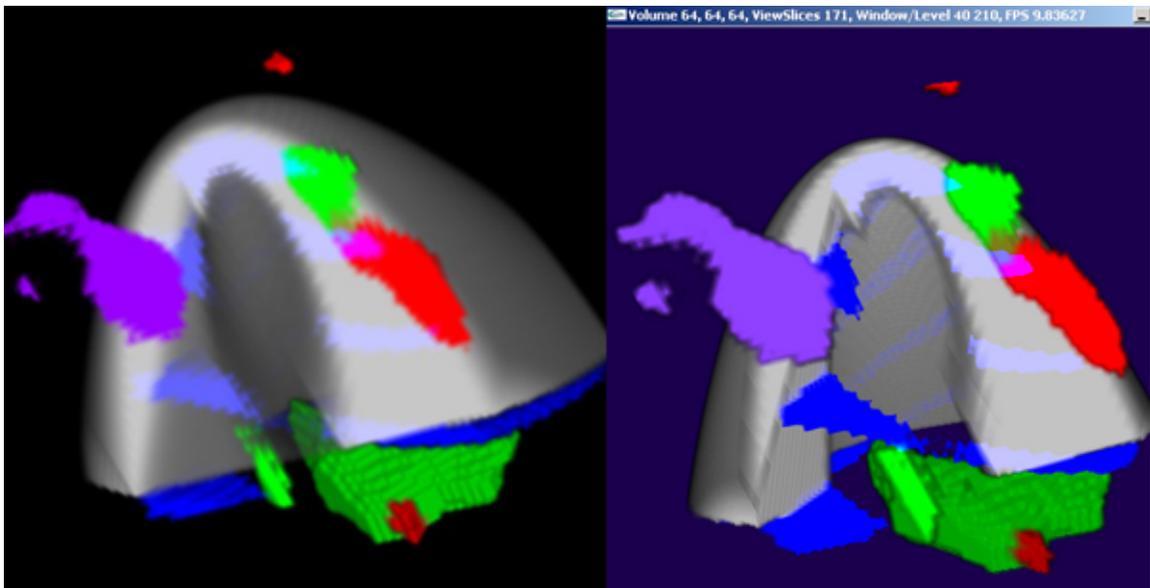


Figure 9: Similar images from the VP1000 (left) and the ATI Radeon with a test dataset.

Displaying Components

To display an activation, we add an opaque white segment to the activation mask for its range and color to the LUT. Otherwise the mask and LUT are zeros and the activation voxels do not contribute. The anatomy and vasculature are easy to turn on and off since they have their own channels.

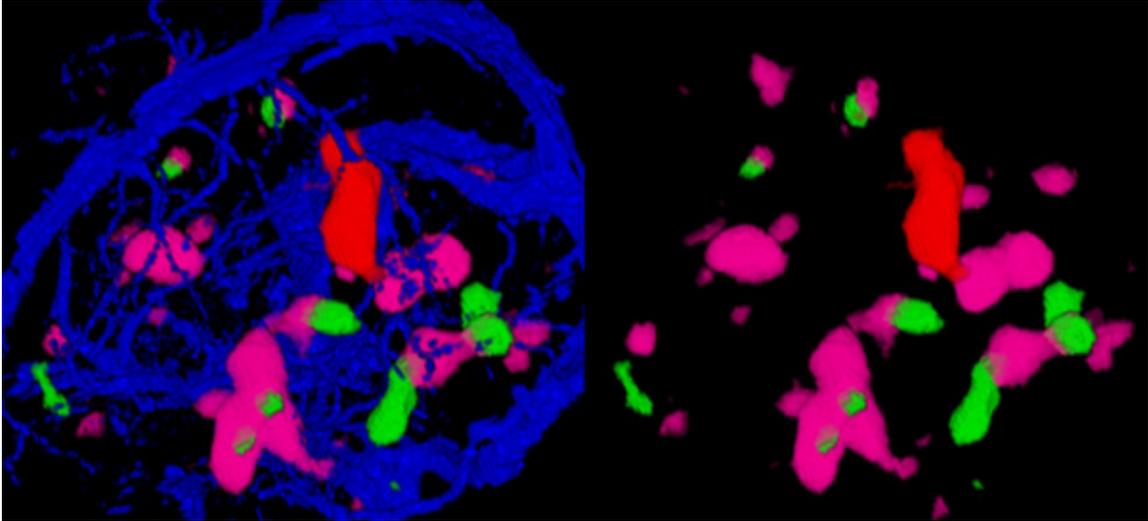


Figure 10: On the left we have all of the activation areas and the vasculature. On the right, just the activation areas are visible.

If the anatomy is visible, we mask it with the activation mask. If it is not, then we mask the vasculature. In all there are 7 options for the VP1000 combiners: a single component (3), each pair of components (3), and all components visible. That would translate into seven different pixel engine programs on the Radeon.

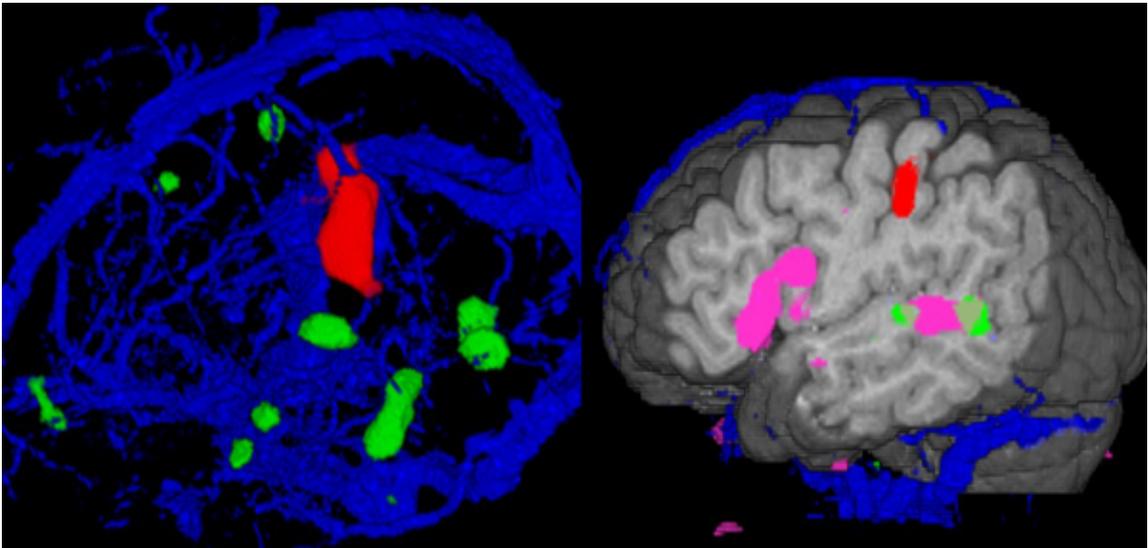
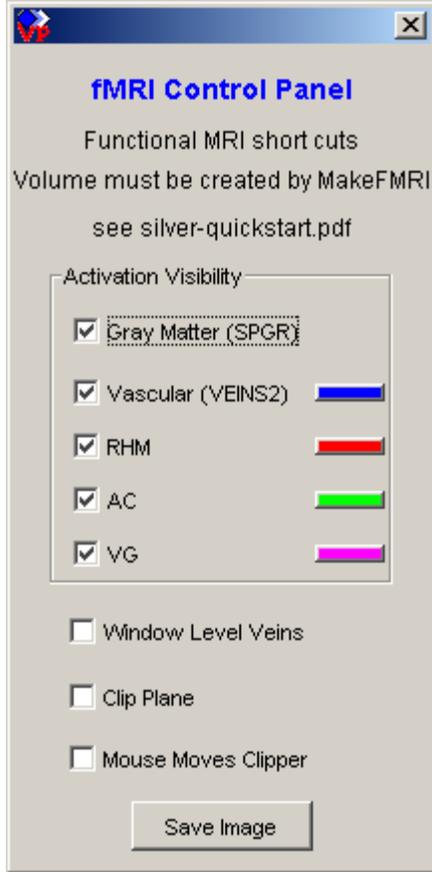


Figure 11: Left, the motor area and the verbal generation area.

fMRI Visualization UI

This is the fMRI Control Panel from our demonstration application.



An fMRI data set that has been created by the MakeFMRIVolume utility will have 3 channels: gray matter (anatomy), vascular and activation areas. These can be independently turned on and off by the check boxes next to each one.

Window Level Veins checkbox toggles back and forth between having the middle button change the opacity of the anatomy or the veins. Window leveling is a very convenient way to control the opacity using a ramp. The window determines how wide the window is. A small window gives a very sharp transition from totally transparent to totally opaque. The level determines the center of the window. A high level shows only the most dense parts of the volume.

Clip Plane turns on the clipping plane half way through the volume and perpendicular to the screen. By default it turns on Mouse Moves Clipper. This make the left mouse button rotate the plane and the right mouse button push it back and forth.

Save image stores the current image in .jpg or .png.

This panel was mostly designed by Shirley Miller of the MGH 3D Imaging Service.

References

COX, RW 1996. AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29: 162-173, 1996.

HIRSCH, J, PH.D.; MAXIMILIAN I. RUGE, M.D.; KARL H.S. KIM, M.D., PH.D.; DENISE D. CORREA, PH.D.; JONATHAN D. VICTOR, M.D., PH.D.; NORMAN R. RELKIN, M.D., PH.D.; DOUGLAS R. LABAR, M.D., PH.D.; GEORGE KROL, M.D.; MARK H. BILSKY, M.D.; MARK M. SOUWEIDANE, M.D.; LISA M. DEANGELIS, M.D.; PHILIP H. GUTIN, M.D 2000. An Integrated Functional Magnetic Resonance Imaging Procedure for Preoperative Mapping of Cortical Areas Associated with Tactile, Motor, Language, and Visual Functions. *Neurosurgery*, Vol 47, No 3.

KNUTSON, B, ANDREW WESTDORP, ERICA KAISER, AND DANIEL HOMMER 2000. fMRI Visualization of Brain Activity during a Monetary Incentive Delay Task, *NeuroImage*, 12, pp. 20-27.

KÖNIG, A, HELMUT DOLEISCH, AND M. EDUARD GRÖLLER 1999, Multiple Views and Magic Mirrors - fMRI Visualization of the Human Brain. Published in the *Proceedings of the Spring Conference on Computer Graphics and its Applications 1999 (SCCG '99)*, Apr. 28 - May 1, 1999, in Budmerice, Slovakia, pp. 130-139.

PIEPER, S , MICHAEL HALLE 1 , RON KIKINIS 2004, 3D SLICER *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2004*; 632-635.

REHM K, STROTHER SC, ANDERSON JR, SCHAPER K, ROTTENBERG DA 1994. Display of merged multimodality brain images using interleaved pixels with independent color scales. *J. Nuclear Medicine*, 35:18-15-1821, 1994.

WU, Y., BHATIA, V., LAUER, H., SEILER, L. 2003. Shear-image ray casting volume rendering. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM Press.

Thank You!

Thanks to Andy Vesper, Yin Wu, T.C. Zhao, Vikram Simha, and Michael Halle for helpful comments. Thank you to Rob Lewis, Shirley Miller, Fresnel Josaphat, Gordon Harris and the MGH 3D Imaging Service for getting me involved and teaching me about fMRI.