

fMRI Visualization of Multiple Functional Areas

Jan Hardenbergh
TeraRecon, Inc.

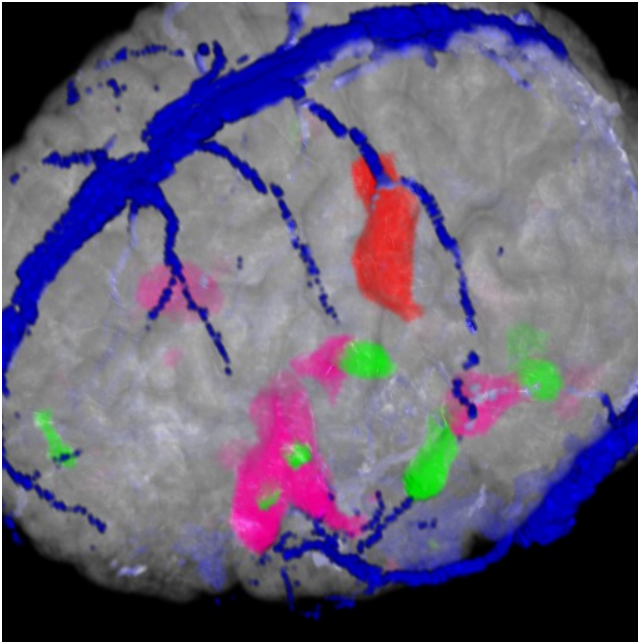


Figure 1. fMRI imaging allows us to see the active areas of brain during mental operations. In this image we see the brain anatomy, the vasculature (blue) and three activation areas: motor (red) and two language (green and magenta).

Introduction

Functional Magnetic Resonance Imaging (fMRI) displays regions of the brain activated for mental functions. Some of the identified functional areas are: language generation and acquisition, motor control, touch, visual processing, and many more. The ability to plan around these areas is very useful in neurosurgical planning.

fMRI takes advantage of the change in blood flow to active regions of the brain. With increased flow, there is a local reduction in oxygen unloaded deoxyhemoglobin, which is paramagnetic. These regions of increase can be identified in the strong magnetic field used for MR imaging.

The reconstructed scan data consists of a volume dataset for each activation area as well as the anatomy and vasculature (blood vessels). We display the anatomy and vasculature to provide context for the activations. In hardware that is two texture reads and two dependent (LUT) reads. Naively, each activation volume requires a texture lookup and two dependent reads. Display of three activation areas requires 13 texture reads, 3 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.

e-mail: jch@terarecon.com

Image rendered the TeraRecon VolumePro 1000. Data courtesy of MGH 3D Imaging Service. Application knowledge: Shirley Miller.

Exposition

Our system uses two techniques, channel masking and activation packing, to view multiple activation areas that do not bleed into the anatomy. Before, four different “channels” (2 activations in context) could be rendered together efficiently on several hardware platforms. With our system, the number of activation areas we can render is limited only by the sum of dynamic ranges.

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. 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.

Without channel masking, colors will blend between channels. When the activations and the anatomy blend, it becomes difficult to identify the activations. 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.

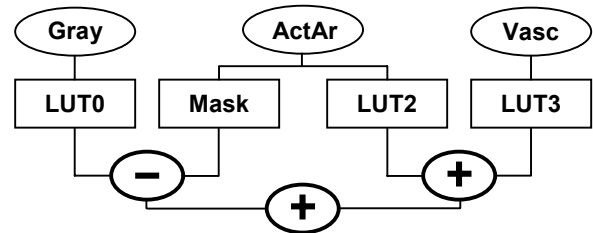


Figure 2. Channel Masking: the RGBA result of the anatomy is masked by the RGBA result of the Mask lookup. The color for the activation areas comes from a second lookup, LUT2.

To display an activation, we add an opaque white segment to the activation mask for its range and color to LUT2. Otherwise, the mask and LUT are zeros and the voxels do not contribute. As the activation ranges do not overlap, they are totally independent. The anatomy and vasculature are easy to turn on and off since they have their own channels.

The system is very interactive on special purpose hardware. It requires only 3 texture lookups and 4 dependent texture reads, and it also works efficiently on commodity graphics hardware.

References

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.