Imaging Brain Activation: The MEG Scalp Projection Tool

Scott A. Carter
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
sacarter@cs.unm.edu

December 10, 2000

Abstract

The recent surge in the extent of neuroscience research has led to an increase in the demand for data visualization methods. The rise of MEG as a means of determining brain activity has led to a particular interest in the spatial representation of data recorded using encephalography. Most of the few software packages capable of doing three-dimensional representations of spatial encephalographic data rely on proprietary software environments. This paper discusses a new method of visualizing data acquired from encephalograph devices using Open Source methods and unencumbered by environmental restrictions.

1 Introduction

In 1886, well before indeed even the notion of computation, William Rudolf Hearst attempted to make a cardiograph of the human brain. Hearst failed in his unprecedented attempt to map the human brain primarily because the skull provided an obstacle too difficult to overcome, but his efforts none the less gave rise to a paradigm shift in the field of medicine. Though the first few attempts to discern the structure of the brain failed, researchers were soon successful in studying the brain via methods such as arteriography and x-rays (Orrison, 1995). Later, arteriography and x-ray exposures became obsolete after research groups developed Magnetic Resonance Imaging (MRI) to measure brain structure and functional MRI (fMRI), Computerized Axial Tomography (CAT), Electroencephalography (EEG) and Positron Emission Tomography (PET) to study brain activity.

The most recently developed method of imaging brain functionality is magnetoencephalography (MEG), a procedure similar to EEG. Both EEG and MEG provide millisecond temporal resolution, which is far superior to other methods, such as fMRI, in which temporal resolution is on the order of minutes. The essential difference between the two encephalographic methods is that while EEG measures electric potentials mitigated by the scalp, the magnetic waves that MEG devices detect exude from the brain unimpeded. To detect the intensity of magnetic activity, MEG systems use an array of Superconductive Quantum Interference Devices (SQUIDS) arranged in pairs. An MEG system may have any number of SQUID pairs arranged either in a plane or, more commonly, around a mould the shape of which approximates a normal human head. EEG, meanwhile, measures intensity of electric activity via a field of electrodes physically attached to the scalp (Papanicolaou, 1998).

While other functional brain imaging methods, such as PET and fMRI, measure brain activation in place, MEG and EEG record brain activation exterior of the brain and skull. Data derived from encephalographic methods, therefore, present visualization considerations. There are essentially two ways of imaging the spatial characteristics of the data: mapping dipole source localizations derived from the data onto MRI images and displaying the data on a representation of the sensor field. Sensor field representations have proven to be especially valuable in providing a means of conducting a first pass analysis of the data. Source localization is a more involved process, requiring more time and effort – reviewing sensor fields a priori can reduce the set of sources to be localized to only relevant sources. More importantly, sensor field representations, or maps, provide intuitive, direct graphical models of encephalographic data.

2 Proprietary Sensor Map Software

A handful of programs exist that are able to display sensor array field maps. One such program, implemented and maintained at Los Alamos National Laboratory, is MRIVIEW, which, while primarily designed for use in viewing and manipulating MRI data, may also be used to visualize MEG sensor intensities (Ranken, 2000). Using MRIVIEW, it is possible to combine three dimensional reconstructions of the patient's head generated from MRI data with an MEG sensor array.

However, while the juxtaposition of the reconstructed head with the field map provides a context for correlation between the actual head and the data recorded, it is not practical to then project the field map directly onto the head since such a transformation would require mutating the data. So, while MRIVIEW provides a well-triangulated and legible field array, the addition of the head reconstruction is visually aesthetic but not entirely useful.

Another potential problem with MRIVIEW is that it is written in a proprietary environment: IDL (IDL, 2000). While IDL is used widely in some scientific communities, programs written in IDL suffer from severe portability limitations. In addition, an IDL site license is prohibitively expensive (a Unix unlimited user node locked license costs $18K) and is time consuming to maintain. Furthermore, IDL has high storage and disk space requirements, and programs run in an IDL environment demand far more system resources than would a stand alone program. Also, MRIVIEW

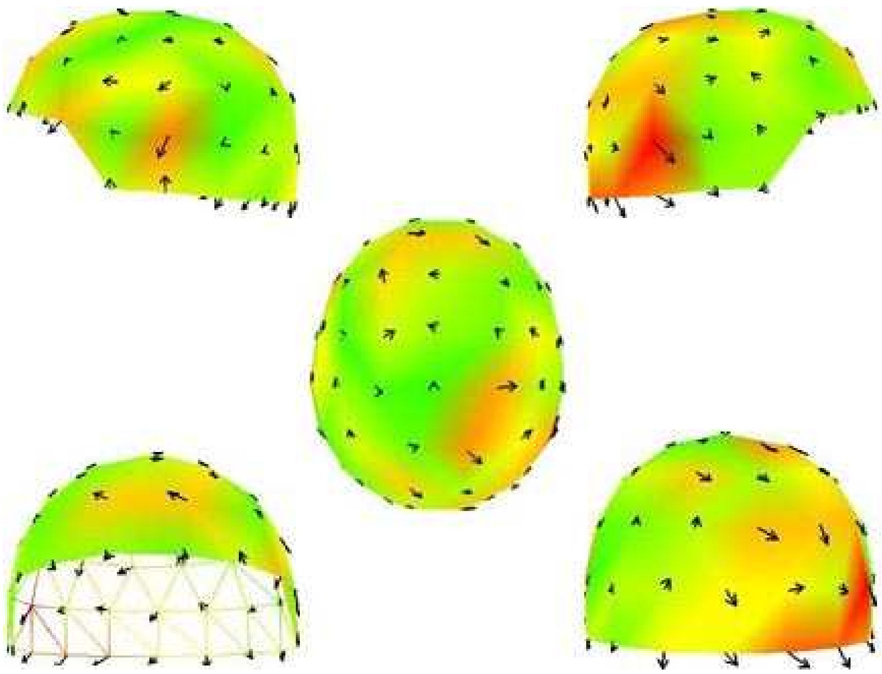

Figure 1: Field maps generated by The MEG Scalp Projection Tool. Shown are the left (top, left), right (top, right), dorsal (center), anterior (bottom, left) and posterior (bottom, right) views of raw, unseparated data at an arbitrary time sample.

requires a considerable amount of human interaction in order to generate a field map from sensor intensity data, limiting one's ability to automate MEG and EEG data processing tasks.

The MATLAB 4D Toolbox for the Analysis of Neuromag Data also provides field map support, though only for MEG (Jensen, 2000). Though MATLAB is more affordable and widely used than IDL, the field map display software in the MATLAB 4D Toolbox, like MRIVIEW, does not lend itself to pipelined meta-applications or image dumps that could be used in, for example, journal or conference papers. Also, MATLAB can exhaust system resources.

3 The MEG Scalp Projection Tool

The inability to easily pipeline the analysis of encephalographic data combined with the monetary and system costs associated with other programs capable of generating field maps motivated the design of The MEG Scalp Projection Tool, a stand alone field map generator developed by Brain Sanchez and the author in the Brain and Computation Lab at UNM (Carter, 1999). Scalp Tool is written in C and uses widely available, au gratis OpenGL libraries and therefore is significantly more portable than other methods. More importantly, Scalp Tool is a stand alone program and supports several flags that may be used to automatically generate image files of a field map.

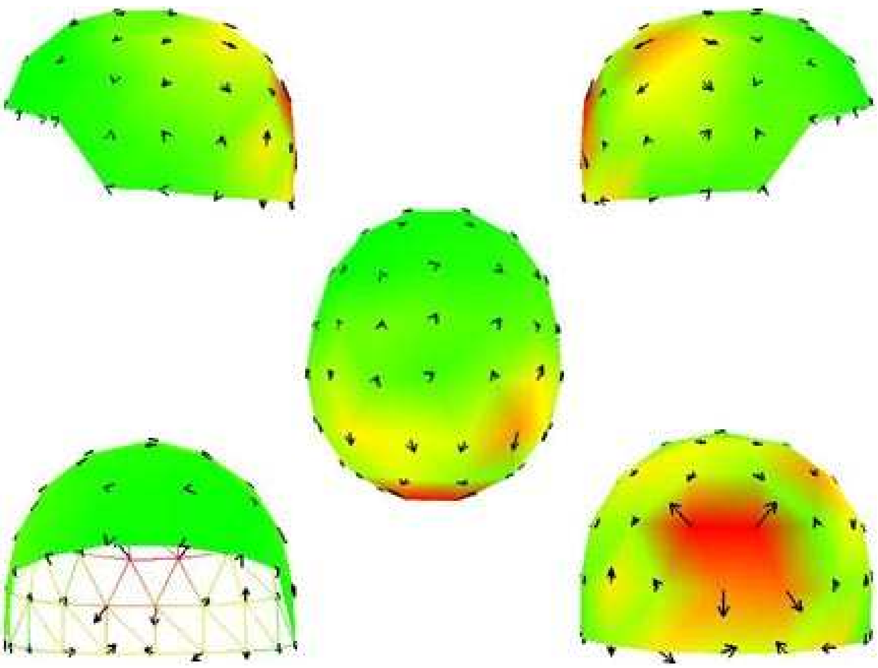

Figure 2: Field maps of a visual source. High sensor intensities in the posterior region indicate occipital activation.

3.1 Functionality

As its name implies, Scalp Tool was originally designed for the display of MEG data, specifically data recorded on an MEG machine using an array of 122 SQUIDS. The program has since evolved and can now handle data collected using other varieties of MEG as well as EEG. As input, Scalp Tool requires only a vector of intensity values – one value for each sensor. If given a series of n-length vectors (where n equals number of sensors), Scalp Tool will display each vector in a continuous loop. In addition, Scalp Tool supports the ability to:

- rotate the sensor field
- show intensity vectors for each sensor (one for each duple of sensors in MEG)
- show text including a title, normalization level and a clock corresponding to the current time frame
- augment or decrease normalization
- overlap two field maps on one array
- zoom

Scalp Tool also supports a variety of colormaps, has means of reading actions from files for presentations and includes window size options.

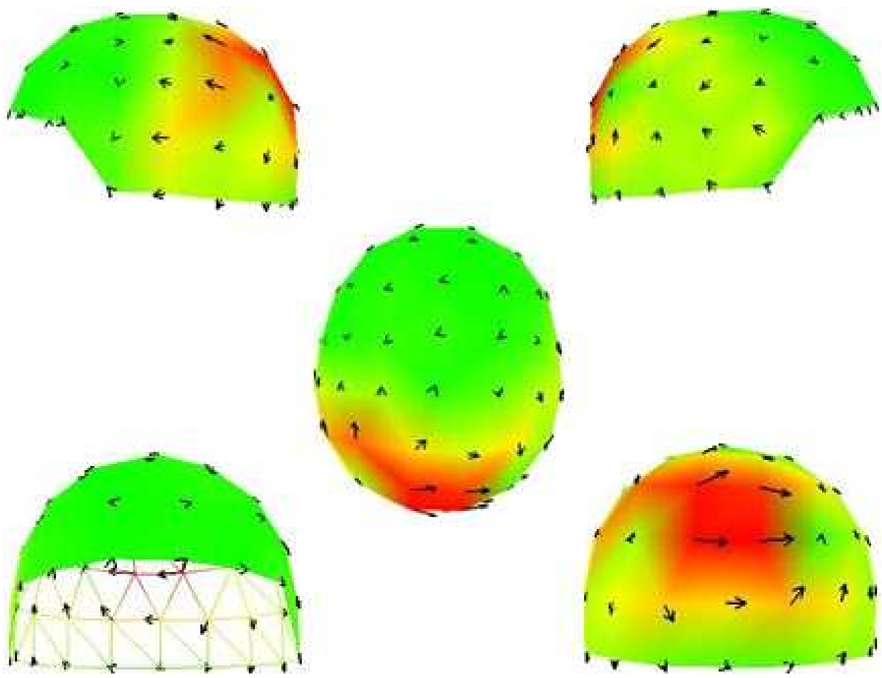

Figure 3: Field maps of a parietal source that shows learning in a classical conditioning protocol.

3.2 Applications

When raw data from an MEG recording is used as input into Scalp Tool, the result is a field map of seemingly random fluctuations that provides little information about the data (Fig. 1). As raw data includes all unfiltered sources, the field map shows a predictable lack of spatial correlation.

What is ultimately desired is a visualization of the different sources that are mixed together in the raw data. The Brain and Computation Lab utilizes Blind Source Separation (BSS) to separate components of raw data. BSS separates measured sensor signals into statistically independent components without requiring a contiguous source model. Also, as BSS uses only temporal information from the sensor signals the algorithm provides no information regarding the spatial geometry of the source, and therefore it is crucial to show the results of the algorithm via a field map. Since BSS is not biased towards any particular field map geometry, the consistency of a components field map with known anatomical organization within the brain provides confirmation that the source component is well separated and meaningful (Tang et al., 2000). Therefore, it is crucial that such a field map be as accurate and precise as the sensor array allows.

The field maps in Figure 2 show views of a separated source in the visual cortex. These field maps show the effectiveness of the BSS algorithm to separate sources and the ability of the Scalp Tool to accurately visualize those sources.

In an ongoing experiment involving neuronal plasticity, Scalp Tool has been particularly useful in the preliminary analysis of BSS separated components of MEG. The protocol involved in this experiment included the presentation of visual and auditory stimuli largely as separate events but occasionally paired together in one event (Carter, 2000). The aim of the experiment is to show classically conditioned responses in brain activation resulting from this occasional pairing. Typically, such learning will only occur in one or two sources, but BSS returns an array of 122 separated components (equal to the number of MEG sensors). Determining that a source shows learning involves generating graphs, calculating statistics and ultimately a manual analysis of all results. However, as putative learning sources should show strong responses in one of a few brain regions, Scalp Tool can be used before more particular data analysis to discern interesting sources (Figure 3).

4 Conclusion

The nineties, "the decade of the brain," saw a surge of research activity in experiments designed to study brain function. The acceptance of MEG as another viable alternative for measuring brain activity has only augmented interest in such experiments. The result is an exponential growth of neurological data that increases the demand for powerful visualization methods capable of coherently displaying large caches of data. Programs such as Scalp Tool have the capability of satisfying this demand.

However, the requirements of neuroscientists will change and so too should Scalp Tool. Studies conducted involving optical recordings of olfaction may benefit from a 3D visualization tool such as Scalp Tool. In addition, it may be possible to port Scalp Tool into a virtual environment and then synthetically to recreate experiments, allowing researchers to see simultaneously both brain activation and the causal stimulus.

Acknowledgments

Dr. Barak Pearlmutter suggested this project originally and not only aided in programming design and implementation issues but also provided both a development environment and funding. Brain Sanchez wrote the OpenGL skeleton of Scalp Tool. Dr. Akaysha Tang provided advice and neurological background. The National Foundation for Functional Brain Imaging provided software and computational support.

References

Carter, S. A. (1999). Online documentation for The MEG Scalp Projection Tool. http://dandruff.cs.unm.edu/~neuro/scalp/.

Carter, S. A., Tang, A. C. Pearlmutter, B. A., Anderson, L. K., Aine C. J., and Christner R. (2000). Coactivation of visual and auditory pathways induces changes in the timing of evoked responses in populations of neurons: an MEG study. In *Society for Neuroscience Abstracts*, 30(9197).

IDL (2000). IDL Product Description http://www.rsinc.com/idl/.

Jensen, O. (2000). Online doctumentation for the MATLAB 4D Toolbox version 1.1. http://boojum.hut.fi/~ojensen/4Dtools/.

Orrison, Jr., W. W., Lewine, J. F., Sanders, J. A., and Hartshorne, M. F. (1995). *Functional Brain Imaging.* Mosby.

Papanicolaou, A. C. (1998). *Fundamentals of Functional Brain Imaging.* Swets & Zeitlinger.

Ranken, D. (2000). Online description of MRIVIEW and associated tools. http://www.isbr.lanl.gov:80/brain_imaging/.

Tang, A. C., Pearlmutter, B. A., Phung D. B., and Carter S. A. (2000). Independent Components of Magnetoencephalography, Part II: Single-Trial Response Time Estimation. In submission.

Appendix – Scalp Tool Documentation

Introduction

The MEG Scalp Projection Tool, or *scalp* is a program designed to render information obtained from a magnetoencephalography device upon a 3D representation of the MEG helmet. Users are likely to run scalp in coordination with other programs developed in the Brian and Computation Lab as part of the DREAMMON suite, including *megtool* and *proj*. However, the program needs only intensity information to operate, and such information may be stored in a specified file.

The DREAMMON suite is licenced under the GNU General Public Licence, or GPL, and is available as a CVS archive. For more information, please contact Professor Barak Pearlmutter (bap@cs.unm.edu).

Using Scalp Tool with megtool and proj

To see a list of possible command line arguments to proj and megtool, type:

```
% proj -help
% megtool -help
```

megtool may be used to pipe intensity information for all channels directly to scalp.

% megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 megdata/meg.raw — scalp -ifile - -no-splash

In the above example, megtool pipes the first 15000 frames of sensor intensity information from megdata/meg.raw to scalp (run "megtool -help" for more information regarding megtool's command line arguments).

In order to view specific data channels, use proj in addition to megtool:

% megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 megdata/meg.raw — proj -W-file Wdata 1 6 — scalp -ifile - -no-splash

Here, proj pipes only channels 1 and 6 to scalp. Any non-zero number of channels may be piped to scalp. Also, it is possible to view two channels independently of one another using scalp's -dual flag.

% megtool -chunk 15000 -data 0 -data-range 122 -dump-channel - megdata/meg.raw — proj -W-file Wdata 1 / 6 — scalp -dual -if - -no-text -no-splash -intensity

When scalp is called with the -dual flag, it expects intensity values alternating between two channels and will display two sets of intensity vectors. Also, scalp will display the first and second sources on green and red color scales respectively. Note that a "/" must be included in the call to proj, indicating that the two channels be piped out in successive frames (i.e., scalp gets frame 1 of 1 then frame 1 of 6 then frame 2 of 1 etc.).

The '#' symbol may be included in the stream of intensity information in order to explicitly show stimulus presentation.

% (megtool -chunk 150 -dump-channel - –data 0 -data-range 122 megdata/meg.raw; echo # foffset -50 p bcolor black#; megtool -chunk 100 -dump-channel - –data 0 -data-range 122 megdata/meg2.raw) — ./scalp -ifile -

The above command line, when executed, will pipe to scalp the first 150 frames from meg.raw, the '#' string and the first 100 frames from meg2.raw. Within the '#' string, the "foffset" flag and its argument, which should be negative, are mandatory and reset the frame number. When the frame number advances to zero, scalp will execute the other arguments in the '#' string. Possible arguments include the bcolor argument and any interactive keystroke. In this case, when the frame number counts up to zero from -50, scalp will set the background color to black and pause (keystroke 'p') animation.

Dumping images to a file

Writing images to a file may be done in one of two ways:

- Click the right mouse button and select the "Write to image file" menu option
- Use the "-dump-image x" command line argument to dump to a file the first x frames scalp encounters

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 200 -no-splash -ifile - -no-text -colormap 6 -bcolor white -image-name test.jpg -dump-image 1 -view R -intensities

Here, scalp will produce an image, "test.jpg", of the first frame of the first channel of meg.raw. Use the -quit flag to close scalp as soon as it has finished writing the image.

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 256 -no-splash -ifile - -no-text -colormap 6 -bcolor white -image-name test.jpg -title "MEG test" -dump-image 1 -view R -intensities -quit

In the above example, the program produces the same image as before, but now closes itself immediately after writing the image file.

Henceforth, all of the examples will include the -quit and -dump-image command line arguments, but note that the image naming conventions remain the same whether the image is dumped at the command line or interactively.

The image name automatically defaults to "title.gif". For example:

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 256 -no-splash -ifile - -no-text -colormap 6 -bcolor white -title "MEG test" -dump-image 1 -view R -intensities -quit

Here, scalp will produce image MEG_text.gif.

If an image name is included in the command line but does not have a recognizable file extension, scalp will write the file in .ppm format.

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 256 -no-splash -ifile - -no-text -colormap 6 -bcolor white -image-name test -title "MEG test" -dump-image 1 -view R -intensities -quit

The resulting output is a .ppm file named "test".

If the "convert" utility is not available on your system, use the -no-convert flag to explicitly dump images in .ppm format.

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 256 -no-splash -ifile - -no-text -colormap 6 -bcolor white -title "MEG test" -dump-image 1 -no-convert -view R -intensities -quit

The resulting output is a .ppm file named MEG_test.ppm1.
The argument to -dump-image may be any integer.

megtool -chunk 15000 -dump-channel - -data 0 -data-range 122 /meg-data/meg.raw — proj -W-file Wdata 1 — scalp -screensize 200 -no-splash -ifile - -no-text -colormap 6 -bcolor white -image-name test.jpg -dump-image 10 -view R -intensities

The above command produces a series of 10 images of the first 10 frames of the first channel of meg.raw.

scalp on SGI

To compile scalp on an SGI machine, go to the scalp/ directory in the DREAMON suite and execute the following command:

% make -f Makefile.sgi

scalp functions properly on SGI machines with one exception – command line image dumping often produces undesirable results. If this occurs on your system, use the interactive pull down menu instead of the -dumpimage command line argument to write images to a file.