Experiences with Matlab and VRML in Functional Neuroimaging Visualizations

Finn Årup Nielsen[1][2] **and Lars Kai Hansen**[1]
[1]Department of Mathematical Modelling, Technical University of Denmark
[2]NEC Research Institute

Abstract

We describe some experiences with Matlab and VRML. We are developing a toolbox for neuroinformatics and describe some of the functionalities we have implemented or will implement and how Matlab and VRML support the implementation.

1. Introduction

Neuroinformatics is the task of organizing, analyzing and presenting neuroscientific data and knowledge [Shepherd et al., 1998; Koslow and Huerta, 1997]. In the branch of neuroinformatics associated with functional neuroimaging ("brain mapping") visualization plays an important role.

Functional neuroimaging based on functional magnetic resonance (fMRI) or positron emission tomography (PET) is a key tool for examination of the living human brain. Brain mapping protocols usually explore the relationship between stimulated cognitive states and activation of specialized brain regions.

fMRI and PET brain scanners can produce sequential volumetric scalar data. Some functional neuroimaging studies involve more complicated designs with several controlled states.

Functional neuroimaging data is seldom viewed directly, but instead it is subject to elaborate statistical processing, the result being one or more scalar volumes, so-called summary statistics. For example activation hot spots can be identified. Center of mass points of "hot spot" areas are tabelized and included in scientific publications with reference to "Talairach-coordinates" [Talairach and Tournoux, 1988]. A database exists with these coordinates: The BrainMap database [Fox and Lancaster, 1994].

The typical visualization task is to display the scalar result volumes sometimes in comparison with other volumes and almost always with some kind of reference, e.g., anatomical images or "Talairach" grid.

We here wish to share our experience on using VRML visualizations for this task and in addition preliminary experiences with Matlab 3D visualization.

We aim at a GUI Matlab program able to display several volumes together and integrating BrainMap data, rendered both in Matlab and in VRML, with possible extended display on virtual reality devices. Both original data, program and the resulting visualizations should be distributable over the Internet.

2. Other Visualizations

Result volumes of a functional neuroimaging study are visualized in a variety of different formats, often supported by tables listing the precise location of the foci center. The formats include gallery of slices with a reference consisting of an anatomical (T1-weighted) MRI image, maximum intensity projection in three orthogonal planes, cortex projected rendering, and "flatmap" where the gyri and sulci of the cortex or cerebellum is flattened producing a 2D representation [Van Essen and Drury, 1997; Hurdal et al., 1999]. Volume rendering is also used, e.g., in AFNI [Cox, 1996; Cox and Hyde, 1997]. The Interface to BrainMap comes either as a web-interface or a stand-alone program [Lancaster et al., 1997], — the web-interface has no visualization.

A user study [Rehm et al., 1998] has shown that in connection with comparison of multiple volumes 3D polygon-based visualization benefits.

For most of the visualization techniques employed in functional neuroimaging visualization it is possible to implement them in either Matlab or VRML. Full volume rendering is not handled in any of the two languages.

There exist several packages for functional neuroimaging visualization and they usually incorporates some kind of analysis as the most important part of the package. For Matlab there are SPM [Friston et al., 1995], and a "fMRI Analysis Package" [Skudlarski, 1998]. A warping package also uses Matlab for visualization [Kjems et al., 1999]. IDL is used in "Corner Cube Environment" [Rehm et al., 1998]. Others are based on compiled programs and usually limited to a specific graphic system, e.g., X for AFNI and Stimulate [Strupp, 1996], X/SGI for CARET

---

[1]Building 321, Lyngby, 2800, Denmark, fn,lkhansen@imm.dtu.dk
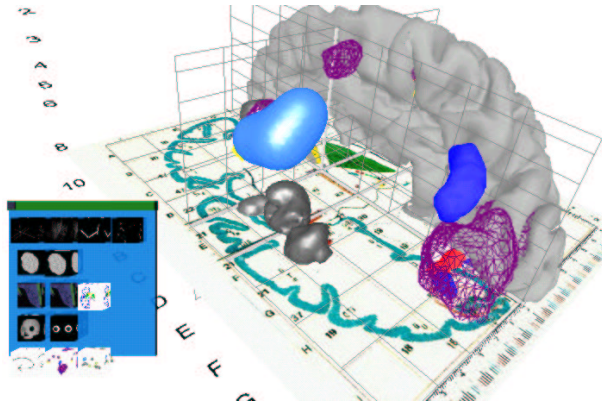[2]4 Independence Way, Princeton, New Jersey, 08540, USA

Figure 1: Screenshot of VRML

[Drury and Van Essen, 1997], or Windows for MRIcro. Few have used VRML/Java [Drury et al., 1997]. All of the packages are distributed freely for research-purposes.

3. VRML

We have used VRML for visualization of functional neuroimaging results. Prior to Matlab 5.3 we used our own program "polyr" for isosurface generation. However, a simple iso-surface program could not suffice for the interactive visualizations we aim for.

Simple control devices (e.g.,. scroll bars etc) can be built, but as VRML, e.g., lacks keyboard interface complex interaction is limited, and CGI-script and java/javascript did not provide a consistent interface. Figure 1 shows a VRML visualization with a rudimentary interface, enabling hide and show of the VRML components.

Our conclusion is that VRML is a scientific visualization *development* environment with serious limitations, and it should serve more as a kind of Postscript/PDF for 3D interactive visualizations.

Virtual reality headset displays with VRML could be obtained, and the VRML-file did not have to be "extended" in any way for it to be displayed within the virtual reality setup. We had access to WorldToolKit, but as the package did not fully integrate VRML and it is certainly not widespread in the functional neuroimaging community we did not pursue it further.

4. Matlab

Matlab is widespread in the functional neuroimaging community partly due to the program SPM.

We have had some experiences with building graphical user interfaces and simple 2D visualizations in connection with our lyngby toolbox.
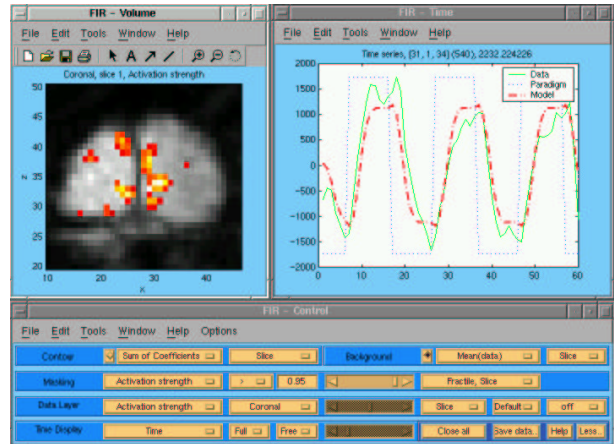

Figure 2: Screenshot of the "lyngby" toolbox.

Our lyngby toolbox [Hansen et al., 1999] is not primarily a visualization environment, but rather built as a "modeler's toolbox", where we have provided several different analysis methods within our pluralistic analysis paradigm [Lange et al., 1999], and we have found Matlab to be a convenient environment for this. We provide a simple 2D visualization for the result of the analysis. Figure 2 shows this interface where a click in the left window changes the content in the right window. The user is able to call the function that constructs the window either from a graphical interface or from the command-line.

The size of the typical functional neuroimaging data set together with Matlab's use of double floating point numbers put a constraint on the design. The other Matlab package SPM uses memory mapping to cope with large data sets, but we chose to use only memory to give us and the user direct access to the central data matrix and ease any extensions. This will of course restrict the size of the data set one is able to work with. It was a serious limitation when we began the development of the lyngby toolbox in 1997, but as RAM-size has increased beyond typical functional neuroimaging data set it is less of a problem.

Prior to version 5 there were no structures in Matlab making it difficult to represent property-like variables. In the lyngby toolbox (which we began with Matlab 4) many global variables were necessary to handle properties. With the structures of Matlab 5 the heterogenous data can be handle more consistently. As BrainMap data is quite heterogenous the visualization environment calls for small-scale database-like capabilities, which has to some extent been honored with Matlab 5.

From version 5.3 Matlab includes some proper 3D visualization functions, e.g., isosurface generation, and data pipes easily to VRML. We have not tried to integrate any virtual reality setup with matlab.

	Matlab	VRML
Color	indexed/RGB	RGB
Scene graph	"flat"	hierarchical
Transparency	no	yes
0-dimensional graphics	yes	(yes)
1-dimensional graphics	yes	(yes)
Volume	yes	(no)
Interactive	yes	partially
Distributable	(yes)	yes
Web-integration	(yes)	yes

Table 1: Comparison between Matlab and VRML

Matlab provides a VRML-generation function where the components of the plot/window are used to construct a VRML file. The function has not been optimal, e.g., there are no means to "plugin" to it, and text and hyperlinks are not translated.

5. Comparison between Matlab and VRML

Working with both Matlab and VRML, we find that none of them are completely optimal for all of our needs. Table 1 sums up some differences between Matlab and VRML.

VRML uses RGB as the color model, and Matlab uses indexed and RGB color. The usual setup of a gray image for anatomical background and a hot image for a masked functional image necessitate the developer to directly control and keep track of the color scale when using indexed color. The RGB-solution is usually more convenient for the developer.

VRML is a hierarchical language where transformations (scaling, translation, ...) can be nested. This approach is more developer-friendly in connection with compounded objects, such as glyphs, where components need to be handled in a group. As Matlab is "flat" the developer will have to manipulate each component directly.

Transparency is not supported in Matlab. VRML-browsers display transparent objects with varying degree of quality. [Rehm et al., 1994] used a poor man's transparency in IDL by interleaving two images. It is desirable to show an activation area at different levels of the threshold. In 2D plot this can be obtained by using a color scale. In 3D without transparency one can use oblique polygons as the top-level and a wireframe at the second level. Using more than one level of wireframes it is not possible to get a clear distinctions between the different layers. Using transparent polygons more levels can be used. However, rendering quality of the transparency in VRML-browsers has not been high enough to make more than one level of transparency useful.

One and zero-dimensional graphics in the form of lines and point graphics (markers) are well supported in Matlab, and it is possible to make lines in VRML but point-style, line-style and thickness are not supported. Furthermore there is no zero-dimensional graphics other than a simple point.

Volumes or data structures in any higher order dimensions are not directly handled in VRML. With Matlab 5 data structures with more than two dimensions became possible. In the lyngby toolbox we vectorized the volume, making it difficult for the user to manipulate slices.

The data is accessible on the command-line in Matlab, whereas in VRML it is hidden in the web-browser. It is very useful with an interpreted language, that does not set a shape boundary between the user and the developer.

It is desirable to have distributable file formats, e.g., the proceedings of human brain mapping conferences are distributed on CD-ROMs, and any visualizations could also be distributed like this. For the usefulness of the distributed file it should be in a format that a wide number is able to read and view. Standardized file formats are more likely to gain acceptance. There exist web-based browsers for visualizations in proprietary formats, e.g., the Sense8 WorldUp browser. Matlab has a visualization distributable format: The figure/plot can be save as a "fig"-file. However, one should not pipe a fig-file to matlab directly from the web-browser, as the callback button-press functions can contain malicious code.

Along with distributability comes web-integration. In VRML hyperlinks can be defined. This is also possible in Matlab as a simple function for calling the web-browser is defined. With this functionality the visualization can be made part of a larger hyperlinked web-presentation [Nielsen and Hansen, 1997; Nielsen and Hansen, 1998].

6. Neuroinformatics toolbox

We are developing a neuroinformatics toolbox. The goal is to construct a tool that the researcher working with functional neuroimaging can use after the actual analysis where she wants to integrate and compare the obtained results with results from other studies. Visualization should be important in this connection. The tool should be flexible so that components can be taken out and used on their own. At the same time we want an interface that enables the user to work solely within a graphical environment.

The following scenario could be an example of a procedure in the toolbox:

1. Load the volumetric result of a functional neu-

roimaging analysis.

2. Display the result in 3D along with reference: Anatomical image, grid and labels among others.

3. Mark a region in the display

4. Inquire the BrainMap database or other database for any other studies (journal articles) reporting brain activations foci in this region.

5. Disregard the foci that are believe to be unimportant.

6. Generate a probability density model for the foci maintained.

7. Display the foci and the associated probability density model together with the resulting volume.

8. Put the result visualization on the web-page together with other information accessible to the whole neuroimaging community.

All these steps are possible in Matlab with varying degree of difficulty and outcome. VRML alone will not support all steps. Most of the steps are not directly related to visualization.

Step one is a trivial procedure but not neglectable as functional neuroimaging is inhabited by a number of different file formats. These file formats are often not standardized formats and thus Matlab does directly supported them. It may be anticipated that the researcher has her own data format and will write her own plugin reading function. Then the toolbox should support callback to this code. In the lyngby toolbox we have such a functionality to support loading of the data for simple cases. This is possible in Matlab as it does not require all the functions to be defined before running. (Matlab checks the function calling, but it is possible to hide the function call behind a "eval" command.) The AFNI package, which is based on C, makes it possible to write plugins.

Step two is the actual visualization task. Figure 3 shows a screenshot from a VRML-browser with a file constructed by a command-line Matlab program. Data from the BrainMap database is displayed together with some "context", e.g., a part of the cortex.

Matlab directly supports region marking in 2D. As all graphic components have callback functions, it is possible to implement step 3 in a variety of different ways. VRML can also have "callbacks" on all graphical objects, though it is hard to get the information "out the browser".

Step four requires the program to download a web-page and parse the HTML. There is no direct Internet
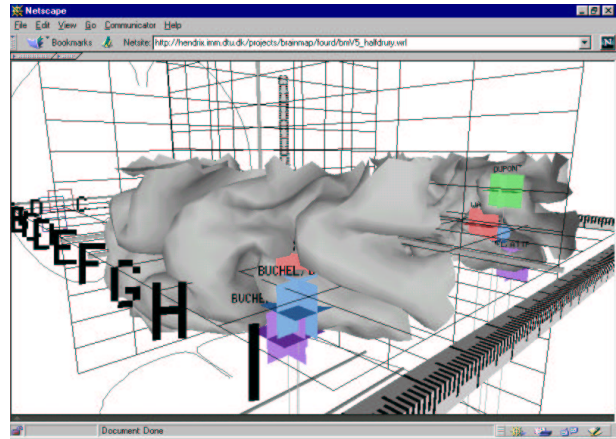

Figure 3: VRML constructed from Matlab

support in Matlab. This kind of functionality either has to be implemented in a compiled library (in Matlab context a so-called "mex-file") or — as we have done — made as a system call, e.g., to perl, wget or telnet/expect. For parsing of text files we found Matlab slow. Small scale parsing where efficiency is not an issue can be done, but any larger scale parsing cannot be made at interactive rate convenient enough for the user.

Step five requires some non-trivial interaction, where support for widgets from the development environment would be advantageous.

Step six — the construction of a probability density volume — is conveniently handled in the matrix language of Matlab.

Step seven requires the modification of the visualization, possibly a whole new generation of it. It is possible to modify the visual components in Matlab.

Step eight is where we find Matlab should not be used, but one should instead resort to VRML.

Many of the steps might be handled in a data flow interaction interface. Functional neuroimaging results are not so voluminous as the original data, so memory-issues will not put a constraint on the design, and a data flow interaction interface with local storage should be possible. Matlab has a data flow environment for simulation in control and signals: "simulink". However, it will probably not be suitable for visualization data flow interfaces. It is not part of the standard matlab distribution and not widespread in the functional neuroimaging community.

7. User experiences

In the two cognitive studies we have been involved in with the lyngby toolbox [Purushotham et al., 1999; Bal-

slev et al., 2000] we have experienced that users often want to use the functions in a slightly different way than we anticipated. Either we or they write a small script that incorporated the new wanted functionality. Though the typical functional neuroimaging researcher will be prepared to write small Matlab scripts, she is not likely to edit VRML files, e.g, to include the small library of VRML components we have made. A GUI interface is necessary (with what humorously has been called "doctor's buttons") before it will become useful for the typical functional neuroimaging researcher.

We strive to make the installation as easy as possible and independent of any third party software. The lyngby toolbox does not require any form of compilation and does not dependent on any other software than matlab. This means that it is possible to download, install and start in less than two minutes. A neuroinformatics toolbox incorporating web-functions will need to rely on compilations or other programs.

8. Conclusion

We find Matlab a convenient platform for development of computational algorithms, and with the visualization functions of Matlab version 5.3 a reasonable amount of visualization techniques are available.

Our experience is that most of the functionality should be accessible from a graphical interface for a toolbox to gain widespread use. At the same time it should flexible enough for the advanced user that would like more than the "predefined interactivity". This is supported by a general purpose interactive (programming) language.

By outputting resulting visualizations to VRML they can be distributed, e.g., on the Internet, and combined in a larger hyperlinked web-presentation.

We will pursue the combination of Matlab and VRML further for our neuroinformatics toolbox.

Acknowledgement

References

Balslev, D., Law, I., Frutiger, S. A., Sidtis, J. J., Nielsen, F. Å., Christiansen, T. B., Strother, S. C., Svarer, C., Rottenberg, D. A., Paulson, O. B., and Hansen, L. K., 2000. "Visuomotor skill learning: a PET study of mirror tracing using cluster analysis and statistical parametric mapping". Submitted to HBM'2000.

Cox, R. W. "AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages". *Computers and Biomedical Research* vol. 29, pp. 162–173, 1996.

Cox, R. W. and Hyde, J. S. "Software tools for analysis and visualization of FMRI data". *NMR in Biomedicine* vol. 10, pp. 171–178, 1997.

Drury, H. A. and Van Essen, D. C., 1997. *CARET User's Guide*. St. Louis, Missouri, USA: Washington University School of Medicine.

Drury, H. A., West, B., and Van Essen, D. C., 1997. *CARETdaemon USER's Guide: Web-based Surface Visualization Software for Atlases of the Cerebral Cortex*. St. Louis, Missouri 63110, USA: Washington University School of Medicine, Department fo Anatomy & Neurobiology.

Fox, P. T. and Lancaster, J. L. "Neuroscience on the net". *Science* vol. 266, no. 5187, pp. 994–996, 1994.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-B., Frith, C. D., and Frackowiak, R. S. J. "Statistical parametric maps in functional imaging: A general linear approach". *Human Brain Mapping* vol. 2, pp. 189–210, 1995.

Hansen, L. K., Nielsen, F. Å., Toft, P., Liptrot, M., Goutte, C., Strother, S. C., Lange, N., Gade, A., Rottenberg, D. A., and Paulson, O. B., 1999. ""lyngby" — a modeler's Matlab toolbox for spatio-temporal analysis of functional neuroimages". See [Rosen et al., 1999], pp. S241.

Hurdal, M. K., Sumners, D. L., Stephenson, K., Bowers, P. L., and Rottenberg, D. A., 1999. "Circlepack: Software for creating quasi-conformal flat maps of the brain". See [Rosen et al., 1999], pp. S250. Fifth International Conference on Functional Mapping of the Human Brain.

Kjems, U., Strother, S. C., Anderson, J. R., and Hansen, L. K., 1999. "A new unix toolbox for non-linear warping of MR brain images applied to a $[^{15}O]$ water PET functional experiment". See [Rosen et al., 1999], pp. S18. Fifth International Conference on Functional Mapping of the Human Brain.

Koslow, S. H. and Huerta, M. F. (Eds.), 1997. *Neuroinformatics, An Overview of the Human Brain Project*. Lawrence Erlbaum Associates.

Lancaster, J. L., Chan, E., Mikiten, S. A., Nguyen, S., and Fox, P. T., 1997. "BrainMap™ search and view". In Friberg, L., Gjedde, A., Holm, S., Lassen, N. A., and Nowak, M. (Eds.), *Third International Conference on Functional Mapping of the Human Brain, NeuroImage*, Volume 5, pp. S634. Academic Press.

Lange, N., Strother, S. C., Anderson, J. R., Nielsen, F. Å., Holmes, A. P., Kolenda, T., Savoy, R., and Hansen, L. K. "Plurality and resemblance in fMRI data analysis". *NeuroImage* vol. 10, no. 3, pp. 282–303, 1999.

Nielsen, F. Å. and Hansen, L. K., 1997. "Interactive information visualization in neuroimaging". In Ebert, D. S. and Nicholas, C. K. (Eds.), *Workshop on New Paradigms in Information Visualization and Manipulation*. ACM Press.

Nielsen, F. Å. and Hansen, L. K., 1998. "Neuroinformatics based on VRML". In Paus, T., Gjedde, A., and Evans, A. (Eds.), *Fourth International Conference on Functional Mapping of the Human Brain. NeuroImage*, Volume 7, pp. S782. Academic Press.

Purushotham, A., Nielsen, F. Å., Hansen, L. K., and Kim, S.-G., 1999. "Separation of motor preparation and execution regions using meta-K-means clustering on fMRI single trial data". See [Rosen et al., 1999], pp. S51. Fifth International Conference on Functional Mapping of the Human Brain.

Rehm, K., Lakshminaryan, K., Frutiger, S., Schaper, K. A., and Summers, D. "A symbolic environment for visualizating activated foci in functional neuroimaging datasets". *Medical Image Analysis* vol. 2, no. 3, pp. 215–226, 1998.

Rehm, K., Strother, S. C., Anderson, J. R., Schaper, K., and Rottenberg, D. A. "Display of merged multimodality brain images using interleaved pixels with independent color scales". *Journal of Nuclear Medicine* vol. 35, no. 11, pp. 1815–1821, 1994.

Rosen, B. R., Seitz, R. J., and Volkmann, J. (Eds.), 1999. *NeuroImage*, Volume 9. Academic Press. Fifth International Conference on Functional Mapping of the Human Brain.

Shepherd, G. M., Mirsky, J. S., Healy, M. D., Singer, M. S., Skoufos, E., Hines, M. S., Nadkarni, P. M., and Miller, P. L. "The Human Brain Project: neuroinformatics tools for integrating, searching and modeling multidisciplinary neuroscience data". *Trends in Neurosciences* vol. 21, no. 11, pp. 460–468, 1998.

Skudlarski, P., 1998. *fMRI Analysis package*. Yale School of Medicine. http://mri.med.yale.edu.

Strupp, J. P., 1996. "Stimulate: A GUI based fMRI analysis software package". In Belliveau, J., Fox, P., Kennedy, D., Rosen, B., and Ungeleider, L. (Eds.), *Second International Conference on Functional Mapping of the Human Brain, NeuroImage*, Volume 3, pp. S607. Academic Press.

Talairach, J. and Tournoux, P., 1988. *Co-planar Stereotaxic Atlas of the Human Brain*. New York: Thieme Medical Publisher Inc.

Van Essen, D. C. and Drury, H. "Structural and functional analyses of human cerebral cortex using a surface-based atlas". *Journal of Neuroscience* vol. 17, no. 18, pp. 7079–7102, 1997.