NeuroLens: an integrated visualization and analysis platform for functional and structural neuroimaging

R. D. Hoge[1], A. Lissot[1]

[1]A.A. Martinos Center for Biomedical Imaging, Charlestown, MA, United States

We have developed an interactive program for image display and analysis using an object-oriented approach, based on the 'Model-View-Controller' (MVC) design pattern [1]. This strategy has facilitated support of multidimensional image data stored in arbitrary file formats. It also allows highly interactive analysis sessions, with rapid user feedback of results, as well as incorporation of interactively designed analysis steps into automated processing pipelines. One of the primary design goals was to provide flexible and immediate display of unprocessed image data, with addition of higher level functionality in a modular, layered fashion using a plugin architecture. The application was implemented in Objective-C for Apple computers based on the G4 (Motorola Inc.) and G5/970 chipsets (International Business Machines) running the MacOSX operating system and using Apple's Cocoa frameworks (formerly NeXTStep).

The core NeuroLens application handles loading, management, and display of multidimensional datasets represented using a simple but carefully designed 'data model' that is file-format independent. Any number of disk storage formats may be supported by subclassing the data model object class and providing custom read methods (currently this approach has been used to support DICOM, MINC, AFNI, COR, bshort, and Analyze formats). Datasets containing two, three, or more dimensions of scalar or vector image values are seamlessly integrated into the data model and visualization layer, while all processing operations are encapsulated outside the core application as plugin modules.

The software supports simultaneous display and superposition of any number of datasets, limited only by system memory. Display of 10 full functional datasets (i.e. all time-points and slices) without performance degradation on G4 systems with 1GB RAM has been routinely feasible. Moreover, there is no dependence on categorization of datasets based on experimental plan (e.g. as functional, anatomical, etc.). Analysis plugins automatically determine whether they can be applied to a particular dataset, based on dimensionality and data type, and can direct output either to a new instance of the visualization environment, allowing interactive investigation of analysis strategies, or to a disk file as part of a scripted pipeline. Processing steps may be performed directly from DICOM files generated by an MR scanner, or from any of the other scientific working formats mentioned above

Through aggressive optimization of analysis modules using the vector processing unit (AltiVec) present on the G4 and G5 chips used in Apple computers, it was possible to accelerate the processing steps required for generation of functional activation maps to the point where motion-correction, 3D spatial smoothing, and generalized linear model (GLM) fitting can be performed on an image series containing 180 time points and 24 slices on a 64x64 matrix in under 30 seconds (dual 2GHz G5, 2GB RAM) with immediate visualization of intermediate results in a concise display format. To compare performance with other analysis implementations and platforms, we wrote an AltiVec-based implementation of the motion correction algorithm used in AFNI [2] and compared execution times to a non-AltiVec (scalar) version on an Apple dual 2GHz PowerPC G5 running MacOSX 10.3.1 and a dual 2.8GHz Pentium Xeon system running RedHat Linux (kernel 2.4.18). Execution times for the reference dataset described above (with moderate motion) were 8.3s (AltiVec/G5), and 13.8s (scalar/Xeon), 17.2s (scalar/G5). Single-threaded versions were used here, to focus on the relationship between processor model and speed; multi-threaded implementations yielded slightly less than a factor of two increase in speed, due to overhead. Increases in speed with the AltiVec-enabled version were significant on older Apple systems; time to align the reference dataset on an 800MHz single-G4 PowerBook using AltiVec-based code was 34s, compared to 54s using the scalar implementation on the same machine.

As proof-of-concept and for assessment of the plugin architecture, we have currently implemented AltiVec-enabled modules for motion-correction, 3D spatial smoothing, GLM analysis, diffusion tensor calculation, maximum-intensity projection, and display of externally generated cortical surfaces. The plugin design is intended to permit development of customized extensions by users, who can implement these using template plugins and Apple's free development tools (Xcode tm). The main program with core plugins can be installed in a single 'drag-and-drop' step as a self-contained installable bundle, with no dependencies on external libraries other than those that are standard on Apple computers.

High-end consumer oriented computers continue to form a major component of the computational infrastructure in many laboratories, due to the combination of high performance and low cost. NeuroLens was designed specifically to capitalize on the capabilities of such systems, including multiple high-speed processors, routine availability of multi-Gigabyte high-bandwidth memory, and single-instruction-multi-data (SIMD) architectures such as the AltiVec. We have also adhered to a strict set of human-computer interface guidelines [3], resulting in an interface that is easy to learn due to its consistency with other widely-used applications. By redesigning the fundamental data management architecture and visualization philosophy, we have been able to demonstrate design concepts that may become important in the next generation of biomedical image processing applications.

References:

1) E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns. Addison-Wesley (1995).
2) R. Cox and A. Jesmanowicz. Real-Time 3D Image Registration for Functional MRI. Magn. Reson. Med. 42:1014 (1999).
3) Apple Human Interface Guidelines, Apple Computer (2003).