AFNI: Software for Analysis and Visualization of Functional Magnetic Resonance Neuroimages

Computers and Biomedical Research, **29**:162-173, 1996.

Robert W. Cox

Biophysics Research Institute
Medical College of Wisconsin
8701 Watertown Plank Road
Milwaukee, WI 53226–0509

e-mail: `rwcox@mcw.edu`

Abstract

A package of computer programs for analysis and visualization of three-dimensional human brain functional magnetic resonance imaging (FMRI) results is described. The software can color overlay neural activation maps onto higher resolution anatomical scans. Slices in each cardinal plane can be viewed simultaneously. Manual placement of markers on anatomical landmarks allows transformation of anatomical and functional scans into stereotaxic (Talairach-Tournoux) coordinates. The techniques for automatically generating transformed functional data sets from manually labeled anatomical data sets are described. Facilities are provided for several types of statistical analyses of multiple 3D functional data sets. The programs are written in ANSI C and Motif 1.2 to run on Unix workstations.

Introduction

In the last few years, magnetic resonance imaging (MRI) has demonstrated the ability to detect changes in cerebral blood volume (*1*), blood flow (*2*), and blood oxygenation (*3, 4, 5*) that occur locally in association with increased neuronal activity. Functional MRI (FMRI) has become one the most highly utilized techniques for noninvasive mapping and analysis of cortical activity in humans (*6*).

With echo-planar imaging and low-inductance local gradient coils, it is possible to gather a single slice of data in under 100 msec (*7, 8*). With 4 mm thick slices, the entire brain can be scanned in under 3 sec. It takes 50–100 scans to produce reliable functional activation maps (*9*). In under 5 minutes a 3D activation map of the whole brain can be acquired. Experimental scanning sessions typically last for 1–2 hours, and produce 10–20 such 3D neuronal activation maps. The volume of data gathered is very large, and it is essential that easy-to-use tools for visualization and analysis of 3D activation maps be available for neuroscience investigators.

Comparison and averaging of neuronal activation data among individuals are complicated by the normal anatomical variability of the human brain. The most widely accepted technique at present for adapting to this variability is to transform 3D activation maps to stereotaxic (Talairach-Tournoux) coordinates (*10*). This transformation relies on identification of a few anatomical landmarks, and is most easily carried out using high resolution (1 mm^3) 3D anatomical data sets. Whole-brain functional MRI data sets are not yet at this resolution, so the best technique for performing the stereotaxic transformation on FMRI volume data is to acquire a high resolution 3D scan at the beginning of a scanning session, determine the transformation for that data set, and then apply that transformation to the immediately following FMRI data sets.

In this paper, we describe the *AFNI* (Analysis of Functional NeuroImages) software package. Our goals in development were (1) visualization, (2) transformation to stereotaxic coordinates, (3) averaging and statistical analysis of multiple functional data sets, and (4) easy to use interactivity. The software was developed in ANSI C, using the Motif 1.2 toolkit. It will run on Unix systems with 8- or 12-bit X11 Pseudocolor displays. The choice of Unix workstations for the computer platform was based on the amount of data and computation needed for functional neuroimaging—the requirements are beyond the present configurations of most personal computers.

Transformation to Stereotaxic Coordinates

Figure 1 illustrates the anatomical landmarks relevant to the Talairach-Tournoux coordinate system: the anterior commissure (AC) and the posterior commissure (PC), two bundles of nerve fibers that connect the cerebral hemispheres across the longitudinal fissure. The anterior-to-posterior (y) horizontal axis of the stereotaxic coordinate system is defined to be tangent to the superior edge of the AC and to the inferior edge of the PC, at the points where they cross the longitudinal fissure. The vertical axis (z) is defined to be perpendicular to the AC–PC line, to lie in the longitudinal fissure, and to be tangent to the posterior edge of the AC. The remaining axis (right-to-left or x) is defined to be perpendicular to the y and z axes. The origin and orientation of the stereotaxic coordinate system have now been defined.

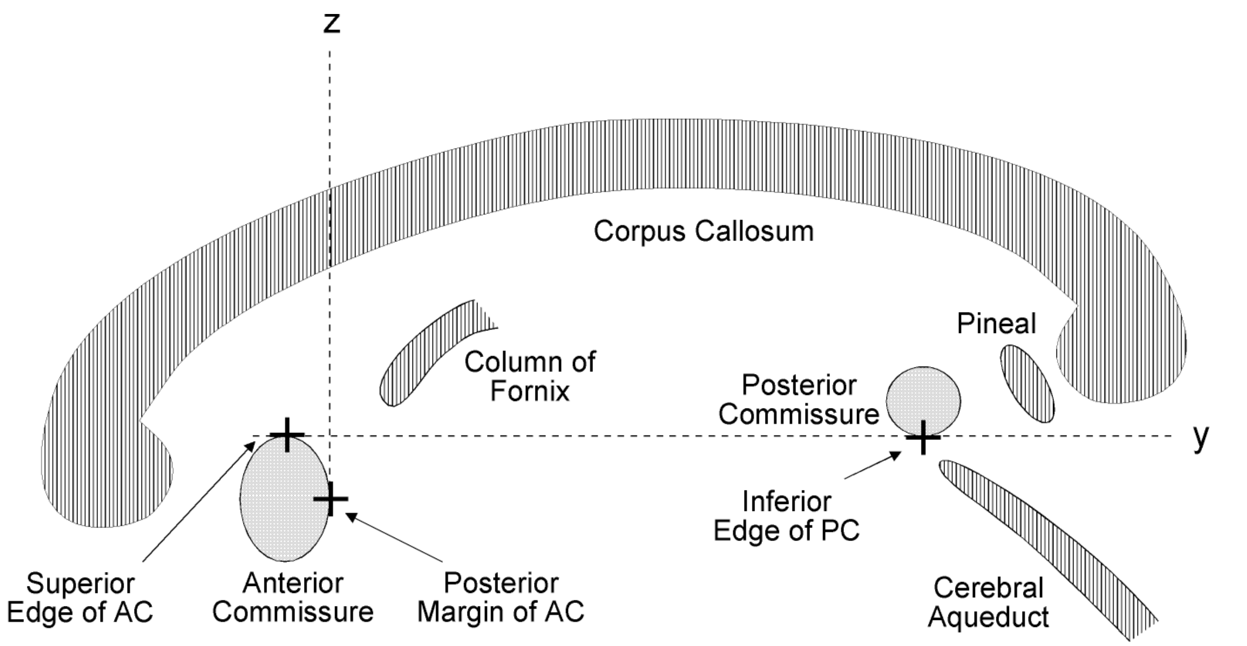

Fig. 1. Sketch of the central portion of the midline of the brain, illustrating the anatomical landmarks used to specify the stereotaxic coordinate system. The Talairach-Tournoux atlas distance from the AC to the PC is 23 mm (*10*).

Normal human brains differ markedly in size. In stereotaxic coordinates, the dimensions along each axis are scaled to the "standard" brain described in the Talairach-Tournoux atlas. The cerebrum is divided into 12 regions: (x) most left to midline, and midline to most right; (y) most anterior (frontal lobe) to AC, AC to PC, and PC to most posterior (occipital lobe); (z) most inferior (temporal lobe) to AC, and AC to most superior (parieto-temporal cortex). The transformation from the original coordinate system in which the data is gathered to the stereotaxic coordinate system is piecewise affine and continuous with 12 patches.

In practice, the transformation is determined in two steps. The first step is to rotate the volume brain image to be aligned with the stereotaxic axes. The second step is to find the extreme points of the cerebrum in the direction of these axes, and then to use those points to scale the volume data set accordingly. The display of a data set in the aligned, but not scaled, coordinate system is called the "AC–PC aligned view" of the given data set. Besides being a step on the way to the "Talairach view," this transformation has its own value. It provides a way to align data sets from the same subject taken on different days without introducing the scaling distortions of the stereotaxic coordinates.

METHODOLOGY FOR TRANSFORMING FUNCTIONAL ACTIVATION MAPS

As mentioned earlier, it is common to gather a whole-brain high resolution anatomical reference data set at the beginning of an FMRI scanning session. With the Spoiled GRASS imaging sequence (*13*), this takes only 10–15 minutes on a clinical scanner. Subsequent functional activation data sets can then be mapped into stereotaxic coordinates once the mapping for the anatomical data set has been determined.

Two practical issues arise in applying the coordinate transformation determined from an anatomical data set to functional data sets. The first complication is that it is common to acquire the functional data sets on a different grid, oriented differently from the anatomical data set. For example, a whole-brain anatomical data set is most rapidly gathered in sagittal planes (since the head is smallest in that dimension), but for many functional purposes, coronal, axial, or oblique acquisitions are preferred (since these planes may line up with the structures being studied). The second complication is that the subject's head may not be in the same location during the functional scan as during the anatomical scan. Related to this issue is the fact that the functional images may be distorted relative to the anatomical images. For example, echo-planar MR images will be warped up to several pixels in the phase-encoding direction by off-resonance effects (*14*). It is possible to map the B_z field and correct for these distortions, but residual errors on the order of $\frac{1}{2}$ pixel are likely. In the context of current technology, this means that 1–2 mm registration errors between echo-planar derived functional images and anatomical images are unavoidable.

We have developed an abstract methodology for keeping track of the transformations required between various data sets so that they can be displayed together. The advantage of this abstraction is that it is not specific to the stereotaxic transformation problem, and so allows for generalization to other situations where geometrical transformations determined from a master data set must be applied to other data sets that are in known geometrical relationships to the master data. The related transformations described below are also a convenient way to organize a computer program.

Figure 2 shows a schematic view of the data sets (boxes) and the "parent-child" relationships between them (arrows) that the symbolic abstractions below are designed to encapsulate. Sans-serif letters such as "a", "b", and "c" are used to denote 3D data sets. The elements of our technique are the following points:

1. Each data set a has associated with it three "parent" data sets (any or all of which can be a itself):

   (a) $\pi_c(\mathsf{a})$ = the coordinate parent, with which it is aligned (point 5c);

   (b) $\pi_t(\mathsf{a})$ = the transform parent, from which it is immediately transformed;

   (c) $\pi_d(\mathsf{a})$ = the data parent, from which it is ultimately transformed (point 5d).

   The functions π_c, π_t, and π_d each take as input a 3D data set and produce as output another 3D data set. In practice, they are implemented as pointers in the C structure that holds a 3D data set. These relations are illustrated in Fig. 2, as applied to anatomical and functional data sets in *AFNI*. The data sets in columns are aligned (coordinate parent-child relation). The data sets at the left of each row are the originals, and the ones to their rights are transformed from them (transform and data parent-child relations).

2. Each data set has associated with it two coordinate systems: a continuous one $\mathbf{r} = (x, y, z)^T$ measured in mm, and a discrete one $\mathbf{q} = (i, j, k)^T$ corresponding to the

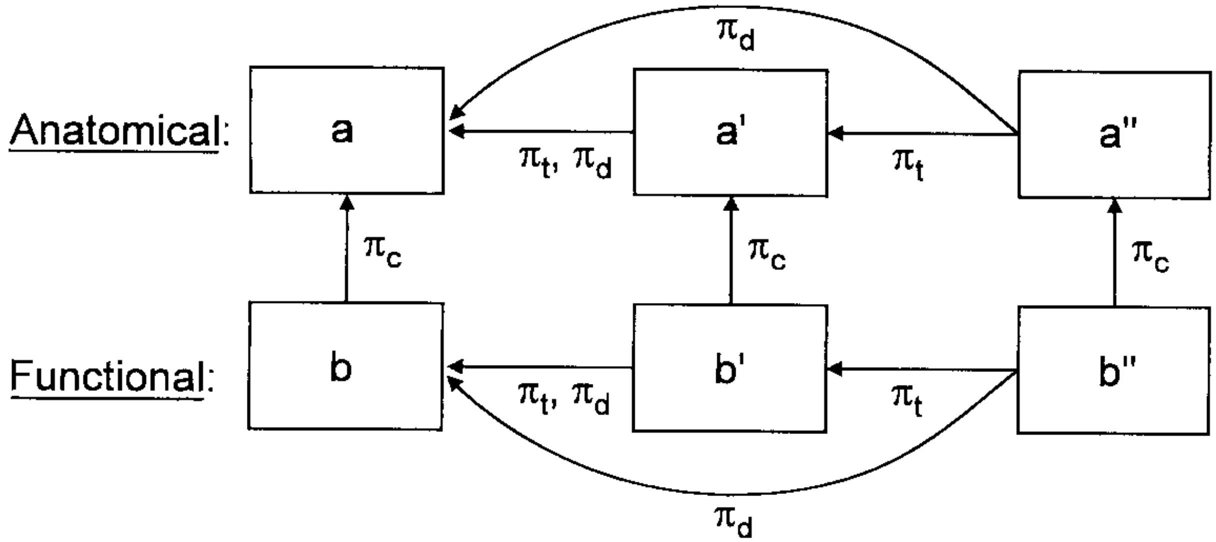

Fig. 2. Arrows show the parent-child relationships among 3D datasets in *AFNI*. Actual data transformations run in the reverse sense of the π_d arrows. Datasets a and b are formed from original data. Dataset a is marked and transformed to a' (AC–PC aligned view) and to a'' (Talairach view). Datasets b' and b'' are automatically generated and transformed from b.

---

underlying voxels. This abstraction allows the transformations to be computed between the continuous coordinates and only at the point of actual application does the mapping to voxels need to be considered (point 5f). In *AFNI*, data sets that are acquired in different orientations will have the same continuous coordinate systems, but different discrete coordinate systems (point 3c).

3. Each data set has three associated geometric transformations between coordinate systems:

   (a) $\mathbf{T}^{\mathsf{a}}_{\pi_t(\mathsf{a})}$, which maps the continuous coordinate system of the transform parent of a into the continuous coordinate system of a. That is, a point whose coordinates are $\mathbf{x}$ in $\pi_t(\mathsf{a})$ has coordinates $\mathbf{T}^{\mathsf{a}}_{\pi_t(\mathsf{a})}(\mathbf{x})$ in a.

   (b) $\mathbf{T}^{\mathsf{a}}_{\pi_d(\mathsf{a})}$, which maps the continuous coordinate system of the data parent of a into the continuous coordinate system of a (point 5e). That is, a point whose coordinates are $\mathbf{x}$ in $\pi_d(\mathsf{a})$ has coordinates $\mathbf{T}^{\mathsf{a}}_{\pi_d(\mathsf{a})}(\mathbf{x})$ in a.

   (c) $\mathbf{D}_{\mathsf{a}}$, which maps the discrete coordinate system of a into the continuous coordinate system of a. That is, a voxel stored at array index $\mathbf{q}$ is at physical position $\mathbf{x} = \mathbf{D}_{\mathsf{a}}(\mathbf{q})$.

4. The set of allowed geometrical transformations between continuous coordinate systems is a nonabelian group under composition ($\circ$), with relations

$$\mathbf{T}^{\mathsf{c}}_{\mathsf{a}} = \mathbf{T}^{\mathsf{c}}_{\mathsf{b}} \circ \mathbf{T}^{\mathsf{b}}_{\mathsf{a}} \qquad \mathbf{T}^{\mathsf{a}}_{\mathsf{c}} = \left[\,\mathbf{T}^{\mathsf{c}}_{\mathsf{a}}\,\right]^{-1} \qquad \mathbf{T}^{\mathsf{a}}_{\mathsf{a}} = \mathbf{I}\,.$$

Specific examples are rotations, scalings, and the piecewise affine transformation to stereotaxic coordinates.

5. For all data sets, the following relationships hold (*cf.* Fig. 2):

   (a) $\pi_d(\mathsf{a}) = \pi_d(\pi_t(\mathsf{a}))$;

   that is, the data parent of a and the data parent of the transform parent of a are the same. This ensures that data sets are always transformed directly from their original sources. In Fig. 2, this relation is exemplified by $\pi_d(\mathsf{a}'') = \mathsf{a}$.

   (b) $\mathbf{T}_{\pi_t(\mathsf{a})}^{\pi_t(\pi_c(\mathsf{a}))} = \mathbf{I}$;

   that is, the transform parent of a and the transform parent of the coordinate parent of a are geometrically aligned. In Fig. 2, the coordinate parent of a dataset is above it, and the transform parent is to the left; for example, $\pi_t(\pi_c(\mathsf{b}'')) = \mathsf{a}'$.

   (c) $\mathbf{T}_{\mathsf{a}}^{\pi_c(\mathsf{a})} = \mathbf{I}$;

   that is, a and its coordinate parent are geometrically aligned. (This is the meaning of what it is to be a coordinate parent.)

   (d) There is a nonnegative integer N such that $\pi_t^N(\mathsf{a}) = \pi_d(\mathsf{a})$, and, if $N > 0$,

   $$\mathbf{T}_{\pi_d(\mathsf{a})}^{\mathsf{a}} = \mathbf{T}_{\pi_t(\mathsf{a})}^{\mathsf{a}} \circ \mathbf{T}_{\pi_t^2(\mathsf{a})}^{\pi_t(\mathsf{a})} \circ \cdots \circ \mathbf{T}_{\pi_t^{N-1}(\mathsf{a})}^{\pi_t^{N-2}(\mathsf{a})} \circ \mathbf{T}_{\pi_t^{N}(\mathsf{a})}^{\pi_t^{N-1}(\mathsf{a})} ;$$

   that is, following the transform parent links back far enough leads to the data parent. The transformation from the data parent to the data set is just the group product of the individual transform parent transformations. In view of these identities, the principal reason for the transform parent relationship to be defined is functional: in practice, one transform can often only be determined after an earlier one has been applied and the resulting data set processed in some fashion. In Fig. 2, a dataset's transform parent is directly to its left, and its data parent is at the far left of its column.

   (e) $\mathbf{T}_{\pi_d(\mathsf{a})}^{\mathsf{a}} = \mathbf{T}_{\pi_t(\pi_c(\mathsf{a}))}^{\pi_c(\mathsf{a})} \circ \mathbf{T}_{\pi_d(\pi_t(\mathsf{a}))}^{\pi_t(\mathsf{a})}$;

   that is, the transformation from the data parent of a to a can be computed as the concatenation of transformations attached to its coordinate and transform parents. This is how "derived" data sets are created (see point 6). In Fig. 2, this relationship means that the transformation from a to a″ is made by composing the transformation from a to a′ with the transformation from a′ to a″. This means that only individual transformations need to be defined by the software, and then the formalism will generate the needed formulas to transform from the ultimate data parent to the final data set's coordinate system.

   (f) The transformation used to map display data elements of a into the data elements that are the original source [in $\pi_d(\mathsf{a})$] is $\mathbf{D}_{\pi_d(\mathsf{a})}^{-1} \circ \mathbf{T}_{\mathsf{a}}^{\pi_d(\mathsf{a})} \circ \mathbf{D}_{\mathsf{a}}$. Interpolation in $\pi_d(\mathsf{a})$ will be needed to faithfully display the transformed data, since in general $\mathbf{D}_{\pi_d(\mathsf{a})}^{-1}(\mathbf{x})$ will not lie exactly on a grid point in the data parent.

6. If a data set b has a coordinate parent $\mathsf{a} \neq \mathsf{b}$, and a in turn has transform children, then b also has transform children derived from a. Assume that $\pi_c(\mathsf{b}) = \mathsf{a}$ and $\pi_t(\mathsf{a}') = \mathsf{a}$. Then the derived child b' of b that corresponds to a' has parents

$$\pi_c(\mathsf{b}') = \mathsf{a}' \qquad \pi_t(\mathsf{b}') = \mathsf{b} \qquad \pi_d(\mathsf{b}') = \pi_d(\mathsf{b}) \; .$$

   In *AFNI*, when the transform child a' is created, the program automatically creates b' as well. If the transformation from a to a' is modified, then the changes are automatically reflected in b'. In Fig. 2, when the user marks the necessary anatomical points in a, then a' can be created. When that is done, b' is automatically created at the same time. Similarly, when the extreme cerebral points in a' are marked, then a'' will be created, and b'' along with it. Combined with the "transform-on-demand" feature (described later), this automatic generation of transformations and transformed data sets means that slices from the anatomical and functional volumes can be viewed in stereotaxic coordinates within a second of placing the anatomical markers.

These points may seem fairly abstruse, but they provide a method for organizing 3D data sets and the geometrical relationships among them as they are created and destroyed by *AFNI*.

Data Organization

Anatomical and functional data in *AFNI* are organized into "datasets." A dataset is a 3D brick of 16-bit values, with some associated information stored in a separate header file. Header information includes the orientation and spacing of the 3D brick voxels, encoded as the transformation $\mathbf{D}_{\mathsf{a}}$ for dataset a. The continuous coordinate system for all datasets is x=right-to-left, y=anterior-to-posterior, and z=inferior-to-superior.

Datasets are labeled as being "anatomical" or "functional." An anatomical dataset contains one value per voxel. A functional dataset can contain one or two values per voxel. The first value is the measured intensity of functional activation. The second value, if present, is a statistical measure of the significance of the activation intensity; for example, the correlation coefficient of Bandettini *et al.* (*9*). The significance value can be used within *AFNI* interactively to threshold the functional display.

Datasets are collected into "sessions" (which are implemented as Unix directories). All the datasets in a session directory are supposed to be associated with each other. In particular, *AFNI* allows any functional dataset from a given session to be color overlaid onto any anatomical dataset from the same session.

Datasets (and sessions) are created with the program *to3d*, which takes as input a stack of images and produces as output a dataset: brick and header files. The user of *to3d* must provide the information necessary to orient the images; for example, the grid spacings and anatomical directions in the images are required inputs. The input images can be blocks of uncompressed 8- or 16-bit data (possibly with header data, which will be skipped).

To3d can be run interactively or in batch mode. The program takes as input contiguous slices of data. In some applications, gaps are present between the functional slices. For example, some investigators using sagittal functional images do not collect data that spans the longitudinal fissure, but instead leave a 3–5 mm gap there. The auxiliary program *abut*

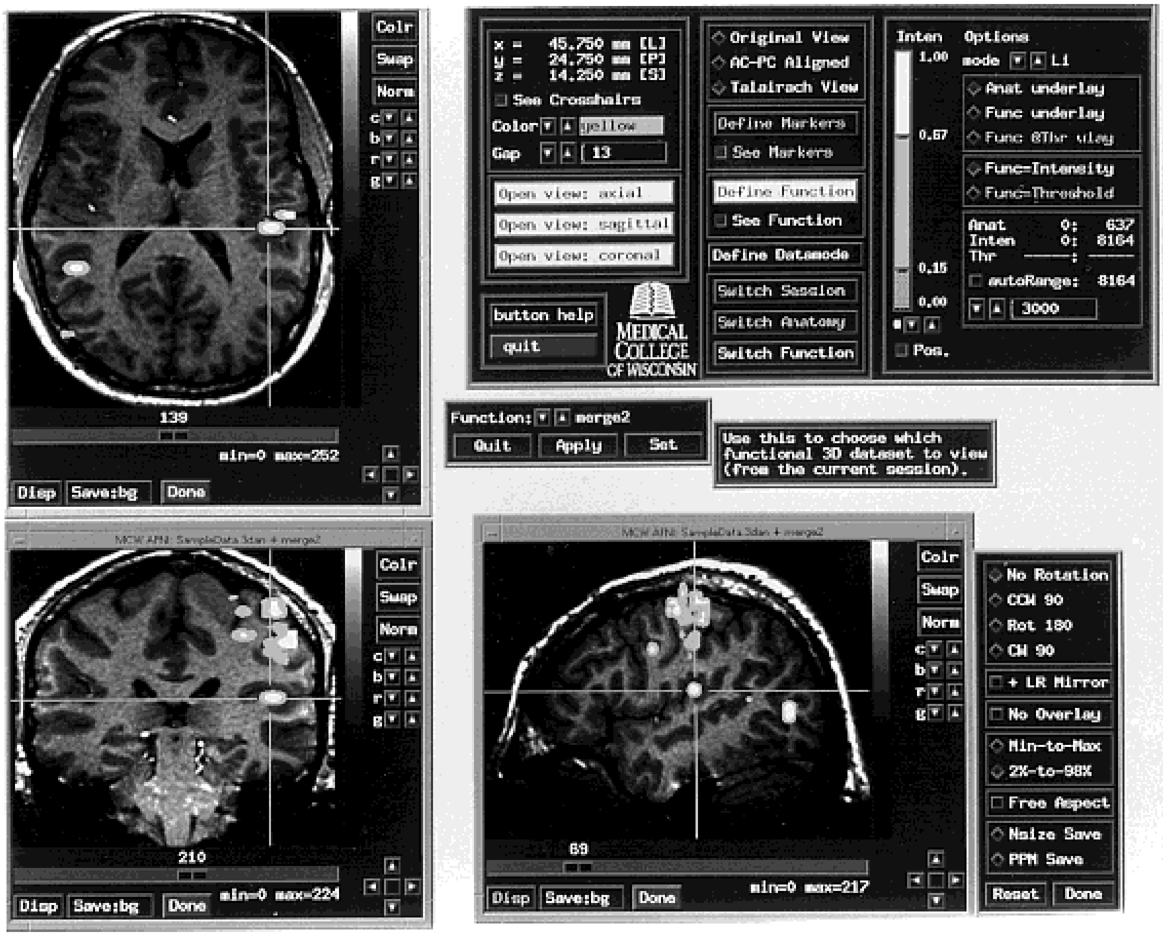

Fig. 3. Screen snapshot from an *AFNI* session, showing the main control panel and the slice image viewing windows.

---

can provide zero-filled images to fill in the gaps, and can resample the nonzero slices to a finer mesh if needed (*e.g.*, to fill a 2 mm gap in a series of 4 mm functional slices). Resampling in the slice select direction between contiguous input slices can be done using nearest-neighbor, linear, or cubic interpolation, at the user's discretion. Interpolation is not done across the boundaries between the gap-filling zero images and the nonzero user-supplied images; that is, interpolation is only done inside spatially continuous blocks of actual input data.

Major Functions of *AFNI* and Auxiliary Codes

AFNI is an interactive X11/Motif program. The user is presented with a main window with buttons and controls for controlling the viewing of images, the transformation to stereotaxic coordinates, and the output of datasets. Figure 3 shows a snapshot of the windows from an *AFNI* session.

IMAGE DISPLAY

Planar images are extracted from 3D datasets along any of the three cardinal (axial, coronal, and sagittal) orientations. Up to three windows for image viewing can be open at any one time, one for each orientation. Colored crosshairs are overlaid in each window, indicating the slice planes that the other windows are using. Controls are provided for moving through the dataset parallel to the coordinate axes. The contrast and brightness of the images can be adjusted interactively.

VIEW MODES AND TRANSFORMATIONS

Each dataset can be viewed in up to three "modes." These are the coordinate systems to which the dataset can be transformed: the original view, the AC–PC aligned view, and the Talairach view. When running *AFNI* for the first time with a given dataset, only the original view is available.

The transformation to the AC–PC aligned view is made by the interactive placement of "markers" on the requisite anatomical landmarks. The exact locations defined by Talairach and Tournoux (*10*) are used: the superior and posterior edges of the AC, and the inferior edge of the PC (*cf.* Fig. 1). To define the vertical axis, the user is also required to mark two points in the longitudinal cerebral fissure. Given the AC–PC line (the new y-axis), one point in this fissure would be sufficient to define the new yz plane. We require two points (separated by at least 20 mm) to provide some redundancy, and to provide some elementary quality controls on the marker set. If the two planes defined by the AC–PC line and the two fissural markers are more than $2°$ apart, *AFNI* will not perform the transformation. The $2°$ tolerance is chosen so that at most a 3 mm alignment error will occur at the edge of the brain. Root-mean-square inter-individual variability in the stereotaxic location of cortical landmarks is approximately 7 mm (*11, 12*). Thus, a 3 mm error in performing the transformation to stereotaxic coordinates does not significantly impact the utility of the transformation for approximately aligning cortical structures among a group of subjects.

The transformation to the Talairach view is made when the user is viewing the data in the AC–PC aligned coordinate system. Six markers must be placed at the extreme points of the cerebrum along each axis. The anatomical landmarks for both transformations are easily seen on Spoiled GRASS 1 mm^3 datasets. With a little practice, it takes only a few moments to place the markers and transform a dataset.

Datasets (functional or anatomical) that are "coordinate children" of an anatomical dataset that has been marked and transformed are automatically transformed along with their parent, as described earlier. *AFNI* does not actually form a new 3D array with the transformed data for each dataset unless ordered to by the user. Instead, the slices that are needed for display are transformed from the original data brick as needed. The user can choose among nearest-neighbor, trilinear, and tricubic interpolation for the resampling method.

This "transform-on-demand" structure has a side-benefit: it is possible to place anatomical markers at sub-voxel locations. This is accomplished in *AFNI* by instructing the program to interpolate the viewed slices to a smaller voxel size; markers are placed at the xyz-coordinates of the display pixel centered in the crosshairs when the user clicks on the "`Set`" button. For example, this technique lets the user "split the difference" when trying to decide if a given pixel or its neighbor is in the mid-sagittal plane. We have used this feature to transform older anatomical data sets, with 4 mm thick sagittal slices and 1 mm in-plane resolution.

FUNCTIONAL OVERLAY

At any given moment, the user of *AFNI* is viewing one anatomical dataset as the grayscale background and (possibly) one functional dataset from the same session as the false-color overlay. The main control window has buttons for switching among datasets and among sessions. Up to 10 sessions with 20 datasets each can be loaded into *AFNI*.

The mapping from functional intensity to colors is controlled interactively. If present, the significance level in the functional dataset can be used to threshold the functional intensity data before display; values below the significance threshold will not be displayed.

The combination of color functional overlay and three orthogonal viewing windows that are easily scrolled has proven to be a very powerful and intuitive mechanism for investigators to understand their functional activation results. The color overlay can be toggled on and off with the press of a button—this allows quick examination of the underlying anatomical structures.

AUXILIARY PROGRAMS FOR MANIPULATING DATASETS

Some functions have not been integrated into *AFNI* itself. These programs operate on 3D datasets, produced either by *to3d* or by *AFNI* when the user saves a transformed dataset to disk. There are four principal auxiliary programs at present:

- *3dclust*: for grouping nonzero voxels into spatial clusters and reporting on their statistics. Clusters can be defined as connected by orthogonal-only adjacency of nonzero voxels (nearest neighbors in the 3D grid), or by allowing connections at diagonals as well (next-nearest and second-next-nearest neighbors).

- *3dmerge*: for editing and combining datasets. This is one mechanism for averaging functional data across subjects after the transformation to stereotaxic coordinates. Editing functions available include discarding voxels not in clusters of a minimum volume (to eliminate isolated points), blurring of voxels to allow for inter-subject anatomic variability (*11, 12*), and discarding voxels that aren't activated (nonzero) more than once in the collection of input 3D functional datasets.

- *3dttest*: for comparing collections of functional datasets. It can perform 1-sample, 2-sample, and paired-sample Student t-tests on the voxel-by-voxel means of 3D datasets. It produces as output a 3D dataset that can be thresholded (within *AFNI* or by *3dmerge*) to display only those voxels which are significantly different from zero (1-sample test) or significantly different between the collections (2-sample tests).

- *3dproject*: for projecting datasets along the three orthogonal directions to produce three 2D image files. This is mainly useful for producing summary images for presentation.

In addition, there are a number of other programs for exploring a functional MRI time series and forming the functional images for input to *to3d*. These programs are described elsewhere (*9*).

Program Design and Implementation

There are three features of the software itself that deserve note. The first has already been introduced: the "transform-on-demand" method of producing images for display on pixel grids that do not align with the original data. This feature has been very useful. For example, the functional datasets do not need to be interpolated to fit the higher resolution of the typical anatomical datasets, but are simply interpolated as needed, even in the "original view." As a practical issue, this saves a great deal of disk space and memory, since a 16-bit $256 \times 256 \times 128$ anatomical dataset occupies 16 Megabytes, while a 16-bit $128 \times 128 \times 24$ functional dataset takes up only 324 Kilobytes. Memory and disk space are also conserved since datasets in the transformed coordinate systems are not explicitly saved (unless requested by the user).

Transform-on-demand also makes the transformation step—after the markers have been placed—almost instantaneous. The actual computations performed are just those to calculate the parameters of the transformation matrices, and to set up the data structures for all the transformed datasets. This occurs so quickly that the user is not aware of all the background work; the only visible indication is that the button to allowing viewing in the new coordinate system becomes usable.

The second noteworthy design feature of *AFNI* is in the code for the image viewing windows. These routines take as input a function to call that will return the n^{th} image in a sequence when n is passed to that function as its parameter. If a 3D brick of data is simply to be viewed directly, the image-supplying function can simply copy data out from the appropriate plane and return it to the viewing window calling routine; such a display is very quick. If the images being viewed must be transformed from an original data brick, then the image-supplying function can call the appropriate interpolation routines. This modularity makes it easy to incorporate the image sequence viewing routines into other programs; for example, *to3d* allows the user to view the images being formed into a 3D dataset, and this is done with the unmodified *AFNI* image viewing routines.

The third noteworthy design feature of *AFNI* is how it accesses datasets on disk. As mentioned earlier, each dataset is stored in two files: a brick file that contains only the 3D data array, and a header file that contains all other information. The header file is pure ASCII, and so can be examined (even edited) by a user. *AFNI* does not actually read into RAM the dataset brick files if they are over a given threshold (200 Kilobytes in the current implementation). Large brick disk files are directly mapped into memory using the Unix `mmap` library function in read-only mode. This allows a disk file to be treated as part of a program's address space, it puts the burden of paging temporarily unused data to disk onto the operating system, and it does not consume system swap space. This is how *AFNI* can run with hundreds of input datasets that far exceed the actual memory capacity of the computer, yet the program has relatively little need to deal with memory management itself. There is, however, one situation where the necessity for allocating and deallocating space for dataset bricks arises. This occurs when the program exceeds the allowable number of open memory mapped files. When this happens, *AFNI* will close the least recently used dataset files in an attempt to allow a new file to be opened.

Conclusions

The software described in this article is a powerful tool for the visualization, transformation, and combination of three dimensional functional neuroimaging results. Its interface has been carefully constructed to be easy to use. Whenever possible, pressing a control "widget" causes an immediate action that is visible to the operator; this rapid feedback makes using the program quite comfortable. If an action takes a significant time to complete (*e.g.*, transforming and writing an entire 16 Megabyte data volume to disk), the program informs the user by changing the cursor to a watch shape.

AFNI is now in daily use by all FMRI researchers at the Medical College of Wisconsin, and has been distributed to over 80 other sites around the world. It is designed to conserve memory and disk space, and so can be run on relatively low cost Unix workstations. The programs are available free (in source code form) for non-commercial research purposes; interested investigators should contact the author.

Acknowledgements

The development of the *AFNI* package was greatly facilitated by feedback and encouragement from many neuroscientists, including J.R. Binder, E.A. DeYoe, J.S. Hyde, S.M. Rao, E.A. Stein, and F.Z. Yetkin. This work was partially supported by the National Institutes of Health grant MH51358.

References

1. Belliveau, J.W., Kennedy, D.N., McKinstry, R.C., Buchbinder, B.R., Weisskoff, R.M., Cohen, M.S., Vivea, J.M., Brady, T.J., and Rosen, B.R. Functional mapping of the human visual cortex by magnetic resonance imaging. *Science* **254**, 716–719 (1991).

2. Kwong, K.K., Belliveau, J.W., Chesler, D.A., Goldberg, I.E., Weisskoff, R.M., Poncelet, B.P., Kennedy, D.N., Hoppel, B.E., Cohen, M.S., Turner, R., Cheng, H.M., Brady, T.J., and Rosen, B.R. Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proc. Natl. Acad. Sci.* **89**, 5675–5679 (1992).

3. Bandettini, P.A., Wong, E.C., Hinks, R.S., Tikofsky, R.S., and Hyde, J.S. Time course EPI of human brain function during task activation. *Magn. Reson. Med.* **25**, 390–397 (1992).

4. Ogawa, S., Lee, T.M., Kay, A.R., and Tank, D.W. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proc. Natl. Acad. Sci.* **87**, 9868–9872 (1990).

5. Ogawa, S., Tank, D.W., Menon, R., Ellerman, J.M., Kim, S., Merkle, H., and Uğurbil, K. Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging. *Proc. Natl. Acad. Sci.* **89**, 5951–5955 (1992).

6. Binder, J.R., and Rao, S.M. Human brain mapping with functional magnetic resonance imaging, *In* "Localization and Neuroimaging in Neuropsychology" (A. Kertesz, Ed.), pp. 185–212., Academic Press, New York, 1994.

7. Mansfield, P. Multi-planar image formation using NMR spin echoes. *J. Phys.* **C10**, L55–58 (1977).

8. Wong, E.C., Bandettini, P.A., and Hyde, J.S. Echo-planar imaging of the human brain using a three axis local gradient coil. *In* "Book of Abstracts, 11th Annual Science Meeting, SMRM, 1992", p. 105.

9. Bandettini, P.A., Jesmanowicz, A., Wong, E.C., and Hyde, J.S. Processing strategies for time-course data sets in functional MRI of the human brain. *Magn. Reson. Med.* **30**, 161–173 (1993).

10. Talairach, J., and Tournoux, P. "Co-Planar Stereotaxic Atlas of the Human Brain.", Thieme Medical Publishers, New York, 1988.

11. Collins, D.L., Neelin, P., Peters, T.M., and Evans, A.C. Automatic 3D intersubject registration of MR volumetric data in standardized Talairach space. *J. Comput. Assist. Tomog.* **18**, 192–205 (1994).

12. Toga, A.W., Jones, A.S., Rothfeld, J.M., Woods, R.P., Payne, B.A., Huang, C., Mazziota, J.C., and Cai, R. Anatomic variability as measured with a 3D reconstructed Talairach atlas. *In* "Quantification of Brain Function, Tracer Kinetics, and Image Analysis" (K. Uemura, Ed.), pp. 459–456., Elsevier Science, New York, 1993.

13. Crawley, A.P., Wood, M.W., and Henkelman, R.M. Elimination of transverse coherences in FLASH MRI. *Magn. Reson. Med.* **8**, 248–260 (1988).

14. Farzaneh, F., Riederer, S.J., and Pelc, N.J. Analysis of T_2 limitations and off resonance effects on spatial resolution and artifacts in echo-planar imaging. *Magn. Reson. Med.* **14**, 123–139 (1988).