Lipsia - A new software system for the evaluation of functional magnetic resonance images of the human brain

Gabriele Lohmann+, Karsten Müller+, Volker Bosch+,
Heiko Mentzel+, Sven Hessler+, Lin Chen*,
D. Yves von Cramon+

+Max-Planck-Institute of Cognitive Neuroscience, Stephanstr. 1a, 04103 Leipzig,
*Max-Planck-Institute of Mathematics in the Sciences, Inselstr. 22, 04103 Leipzig

1 Introduction

Functional magnetic resonance imaging (fMRI) allows to create digital images that display local changes in blood flow with a spatial resolution of about 3 mm and a temporal resolution of less than 1 second. Since its invention a few years ago [1], it has become one of the most important technologies used in human brain mapping research.

The data produced by a typical fMRI experiment consist of a time sequence of digital images taken every n seconds with $n = 2$ being a typical choice. Images consist of several 2D image slices. Usually, two or more experimental conditions are alternated within the same experiment. For instance, some baseline condition may be contrasted with a condition in which some visual or auditory stimulus is presented.

The analysis of fMRI data comprises several aspects. The most important objective is to obtain a statistical parametric map that depicts areas within a subject's brain that show a statistically significant response to an experimental stimulus. In order to obtain such a map, some preprocessing steps as well as elaborate statistical evaluations are needed.

In addition to producing activation maps for individual subjects, methods of inter-subject comparisons are needed so that group studies involving several subjects become possible. This in turn requires methods of spatial normalization and coregistration where data acquired from different subjects are geometrically aligned. Lastly, the results obtained from the statistical analysis must be visualized graphically so that they can be presented to the scientific community.

This article describes the various aspects of fMRI data analysis as implemented in a new noncommercial software package called Lipsia (Leipzig Image Processing and Statistical Inference Algorithms). The Lipsia software was designed to enable the routine processing of large numbers of large data sets. Lipsia currently functions in an environment in which fMRI experiments are conducted for about 10 hours daily generating about 2 Gigabyte of data every day. As new imaging hardware becomes available that allows greater spatial resolution this number may increase by a factor of 2 in the near future.

Lipsia was designed with the following objectives in mind. Clearly, it must be able to handle such large amounts of data efficiently. Furthermore, as the data processing is performed by scientific personell such as psychologists or medical doctors who are are neither experts in computer science nor in image processing, the software must be easy to use and robust. Another very important objective was to support visualizations of individual brain anatomy so that experts can assess the exact location of activation patterns in individual brains.

This article is organized as follows. We begin by describing the various types of data pertaining to an fMRI experiment. We then explain the processing chain that these data are subjected to. Algorithms that are described elsewhere in the literature are briefly summarized. New algorithms are described in more detail.

2 Data types

In the course of an fMRI experiment, three or sometimes four different types of data are generated.

The first data type are the functional data consisting of a time series of scans where each scan consists of a stack of 2D parallel image slices. The most common imaging technique is called Echo Planar Imaging (EPI). In EPI, only one slice can be acquired at a time. Typical imaging parameters are an acquisition time of 125 ms per slice, with 16 slices constituting one image scan. The image matrix usually contains 64×64 voxels that have a spatial with-plane resolution of 3×3 mm with an inter-slice distance of 2 mm and a slice thickness of 5mm. Other acquisition parameters are of course also possible. An EPI time series may contain up to several thousand time steps.

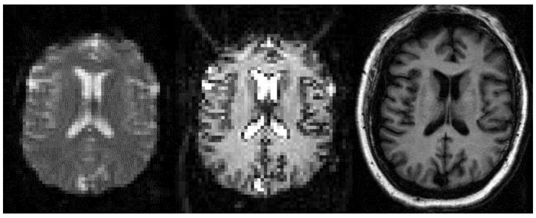


Figure 1. Examples of data types (left to right): EPI, EPI-T1, MDEFT

The second type of data are T1-weighted anatomical scans. They are geometrically aligned with the functional data and are acquired in the same session. They do not contain functional information. Their primary use is to provide an anatomical reference for the functional data. In subsequent sections, we will refer to these data as the *'2D structural slices'*.

2D structural slices can be generated using several different puls sequences. The Lipsia software has been tested with data generated by the MDEFT and by EPI-T1 puls sequences. MDEFT (Modified Driven Equilibrium Fourier Transform) images have a higher within-plane resolution of 0.75×0.75 mm and a better image contrast. EPI-T1 data on the other hand are useful for bridging the gap between functional and anatomical data types, because they are geometrically closer to the functional data and radiometrically closer to the high-resolution MDEFT data.

Lastly, a 3D high-resolution T1-weighted data set is acquired once for each test subject. It is usually not acquired in the same session as the functional data. This data set is used as an anatomical reference for spatial registration and normalization. We will refer to this data type as the *'3D reference image'*.

3 Algorithms

3.1 Preprocessing

As noted above, fMRI slices are usually not acquired simultaneously. To correct for the temporal offset between the slices acquired in one scan, Lipsia offers both a linear interpolation and a sinc-interpolation algorithm. The sinc-interpolation algorithm is based on the Nyquist-Shannon-Theorem.

Baseline drifts are eliminated by a highpass-filter. The cutoff-frequency is specified by the user. As an alternative, Lipsia also offers a highpass-filter as part of the statistical modeling.

3.2 Spatial transformations

In order to perform multi-subject studies, the fMRI data must be rotated and scaled such that the data sets of all subjects are geometrically aligned with each other. This is achieved in two steps: co-registration and spatial normalization.

For multi-subject statistics, a standard stereotactic coordinate system is required. An international convention is the so-called 'Talairach space' [2]. In Talairach space, the origin of the coordinate system is the upper boundary of the anterior commissure (AC). The x-axis is defined as a straight line through the upper boundary of AC and the lower boundary of the posterior commissure (PC). A horizontal line through AC, perpendicular to AC-PC defines the y-axis. Finally, a straight line perpendicular to both the x- and the y-axis yields the z-axis. The Lipsia -software adheres to these conventions.

The 3D reference data that are used as a spatial reference in co-registration must reside in the stereotactic coordinate system. A method for transforming data into this system is described in [3].

3.2.1 Co-registration

Co-registration is cast into an optimization problem: we aim at finding the 6 parameters (3 rotational and 3 translational) that yield an optimal match between the 3D reference data set and the rotated and shifted 2D structural slices. The matching metric is defined by the voxel-wise linear correlation coefficient. Other matching metrics such as mutual information could have been used as well, but since the grey value contrast between the two data sets is quite similar we found that best results were obtained with a simple linear correlation approach.

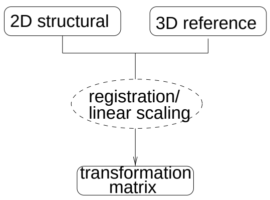


Figure 2. Co-registration.

The strategy for finding the optimal transformation parameters is the following. From the data acquision procedure we have a rough knowledge of the slice orientation. We know whether the orientation is axial, coronal or sagittal, or in some other arbitrary orientation. This initial information is used as a starting point in the search procedure. We first perform an exhaustive search in the vicinity of the starting parameters using a relatively coarse search grid. This step yields a first approximation to the transformation parameters.

This approximation is used as a starting point for an iterative multi-dimensional search procedure called "Powell's Direction Set Method" ([4, pp.412-420]). Powell's method finds the local maximum of a function that is closest to the starting vector. As the starting vector is found by a careful exhaustive search, it is ensured that the resulting vector will be a global maximum. The computing time is about 5 minutes on an SGI Unix-workstation.

The resulting transformation parameters can now be used to transform either the 2D structural slices, the functional 2D slices or a statistical parametric map using a trilinear resampling method.

The registration accuracy can be visually assessed by comparing the location of easily identifiable landmark points in both data sets. After registration, these landmarks should coincide.

The above procedure has by now been applied to several hundred data sets and has proven to be extremely robust. It also works for partial data sets containing only a few 2D slices. It has been tested for both high-resolution 2D structural slices as well as for EPI-T1 data.

3.2.2 Normalization

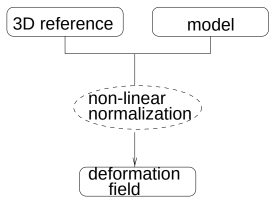


Figure 3. Non-linear normalization.

Lipsia offers a linear and a non-linear normalization. In linear normalization, the data set is linearly scaled to a standard size. In non-linear normalization, an anatomical 3D data set is deformed such that it matches another 3D anatomical data set that serves as a model image. The deformation field obtained by this procedure can also be applied to a statistical parametric map provided it is geometrically aligned with the structural data set that was used to compute the deformation field. The algorithm implemented in Lipsia is described in [5].

3.2.3 Motion correction

The aim of motion correction is to realign the images acquired in an fMRI time series such that they are geometrically compatible. Misalignments occur when subjects move during the course of the experiment. As experiments may sometimes last more than 40 minutes, some form of motion is usually inevitable.

In motion correction, the scans of an fMRI time series are geometrically rotated and shifted until they match a reference scan. In LIPSIA, a matching metric based on linear correlation is used. As for registration, we use Powell's Direction Set Method [4, pp.412-420] to find optimal motion parameters.

3.3 Statistical analysis

The statistical evaluation in LIPSIA is based on algorithms introduced by Friston et al. [6, 7, 8, 9]. In the following, we briefly describe the most important aspects of these algorithms. A more detailed information is given in the references cited above and in [10].

3.3.1 Analysis of individual data sets

The general linear model (GLM) forms the basis of the statistical evaluation in LIPSIA. The GLM has the form [11]:

$$\mathbf{y} = \mathbf{X}\beta \ + \varepsilon \tag{1}$$

where $\mathbf{y}$ represents the vector of measured data which in our context is an fMRI time course containing n timesteps as measured in one voxel. The matrix $\mathbf{X}$ encodes the experimental design. If the experiment contains m conditions then $\mathbf{X}$ is an $n \times m$ matrix where each column encodes the time course of one condition, e.g. if $x_{ij} = 0$, then the condition j is not active at time step i. The m-vector β is unknown and will be estimated in the course of the statistical analysis. The vector ε denotes the error term.

Equation 1 is solved for the unknown vector β using the Moore-Penrose inverse $\mathbf{X}^+$ of $\mathbf{X}$ such that

$$\hat{\beta} \ = \ \mathbf{X}^+\mathbf{y}. \tag{2}$$

Note that the GLM presupposes that the vector components of $\mathbf{y}$ are stochastically independent. Due to the inertia of blood flow this is usually an unrealistic assumption to make. To correct for correlations between scans, the degrees of freedom are reduced using the approach described in [12, 13].

The design matrix $\mathbf{X}$ is chosen such that it reflects not simply the on/off states of experimental conditions. Rather, it models the hemodynamic response expected when conditions becomes active or inactive. The hemodynamic response is known to lag behind the onset of a stimulus such that the time to peak is about 6 seconds. The simplest form of hemodynamic modelling is achieved by convolving the design matrix X with some suitable kernel K such that $G = KX$. Equation 2 then modifies to:

$$\hat{\beta} \ = \ \mathbf{G}^+\mathbf{K}\mathbf{y}.$$

LIPSIA supports two different types of experimental designs that can be distinguished by the duration of the experimental stimuli. The first type is characterized by short, pulse-like stimuli, the second type contains stimuli that have a longer duration. The hemodynamic modeling expressed by the design matrix $\mathbf{X}$ is different for these two types. Hemodynamic modeling in LIPSIA follows the approaches described in [6, 7, 8, 9].

Once the regression coefficients $\hat{\beta}$ have been estimated using equation 2, a statistical inference analysis is performed in which the significance of experimental effects is tested. The weight of the effects is expressed in a *contrast vector* $\mathbf{c}$. For example, a contrast vector $\mathbf{c} = (-1, 1)$ puts the first experimental condition into contrast with the second condition.

In order to perform a statistical inference analysis, a t-statistic is computed that yields a parametric map of significance values based on the null hypothesis that the effects are not significant. The t-value is a measure of the significance of a contrast:

$$t = \mathbf{c}\hat{\beta}(\varepsilon^2\mathbf{c}\mathbf{G}^+\mathbf{K}(\mathbf{c}\mathbf{G}^+\mathbf{K})^T)^{-\frac{1}{2}} \tag{3}$$

where ε^2 is an estimation of the variance σ^2 and can be computed by using the error term of the GLM. The t-values are then normalized into z-values.

In LIPSIA, there are two types of output of a statistical evaluation. The first type is a so-called *z-map*. In a z-map, each pixel contains the normalized statistical significance value that is derived from normalizing the t-value as given in equation 3. The second type of output are so-called *contrast maps*. In a contrast map, each pixel encodes a contrast γ that is defined as $\gamma =$

$c\hat{\beta}$. The contrast is not a measure of the statistical significance but rather a measure of the effect strength.

3.3.2 Analysis of groups of data sets

At the second level of analysis, the z-maps or contrast maps obtained for individual test subjects are subjected to further statistical tests that allow multi-subject comparisons.

In order to test the hypothesis whether the mean contrast across a group of subjects is different from zero at a given voxel, a one-sample t-test is performed with the t-value being:

$$t_c = \frac{\overline{c}}{\hat{\sigma}_c}\sqrt{n},$$

where $\overline{c}$ denotes the mean of the contrasts of the n subjects, and $\hat{\sigma}_c$ denotes the estimated standard deviation. t_c is t-distributed with $n-1$ degrees of freedom. The resulting t_c-values are subsequently transformed into z-values and stored in a z-map.

Likewise, a one-sample t-test can be performed that tests whether the mean z-value is different from zero across a group of subjects [14].

In some experiments, it is of interest to determine whether an activation pattern is significantly lateralized, i.e. whether the activation is significantly stronger in one hemisphere compared to the other. In such cases, user-defined sphere-shaped regions of interest (ROI) centered on the activation areas are used. For each ROI, a mean contrast or z-value is computed, and entered into a second level ANOVA [14].

3.4 Visualization

Cross-sectional slices can be visualized in LIPSIA using a mouse-driven graphical user interface. Individual voxels are selected via mouse-clicks to display time courses of functional data. Images of several individual subjects can be displayed side by side in the same viewer and geometrically linked so that inter-individual comparisons are possible.

Figure 9 shows an example. The rightmost part of figure 9 depicts a line representation of the cortical folds of this individual test subject. In the following, we briefly describe how this representation was derived, beginning with the description of a white matter segmentation algorithm.

3.4.1 White matter segmentation

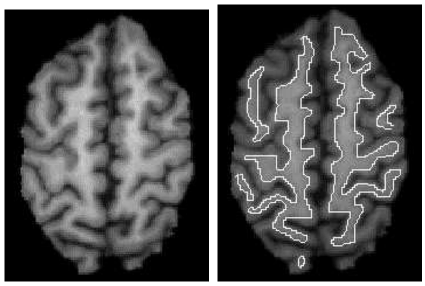


Figure 4. White matter segmentation. The left image shows the original image. In the right image, the white matter boundary is superimposed.

The human brain contains a grey matter and a white matter compartment. In a white matter segmentation, the grey matter and non-brain tissue is removed so that the cortical indentations become more pronounced. Manual segmentations of white matter are not feasible as they are too time-consuming and also affected by operator bias. Numerous automatic segmentation algorithms have been reported in the literature. For a review see for instance [15],[16].

The simplest automatic approaches are based on various forms of thresholding. However, these techniques often do not give satisfactory results due to inhomogeneities in the magnetic field of the scanner. These inhomogeneities lead to significant variations in the grey level distributions across the image so that white matter voxels may occupy a wide range of grey levels that overlaps with other object classes. As a result, global thresholds are usually not feasible.

Scanner inhomogeneities in MR images cause multiplicative low-frequency distortions of the image which is sometimes called the 'bias field' ([17]). The bias field is often a major impediment to reliable segmentation, and therefore this problem has been specifically addressed in a number of research efforts (e.g. [18],[17]).

The algorithm we propose here differs in that it circumvents the inhomogeneity problem rather than trying to solve it directly. The main idea is to use anatomical knowledge that provides sufficient spatial constraints so that the grey/white matter boundary can be estimated regardless of grey level inhomogeneities. Specifically, we exploit the fact that the brain's white matter is covered by layers of grey matter which can be iteratively eroded until the white matter surface appears (see figure 4).

The method combines region growing with techniques known from topological thinning and hysteresis thresholding. The algorithm begins by attributing a depth label to each foreground voxel where a voxel is considered a foreground voxel if it represents brain tissue. Depth labeling can be achieved using a 3D distance transform [19]. A distance transform attaches a label to each foreground voxel that encodes the length of the shortest path towards the closest background voxel.

We then collect all depth-labeled foreground voxels into an array and sort the array by depth. The algorithm proceeds by deleting successively deeper voxels from the array in a manner similar to 3D topological thinning ([20]) until the boundary between grey and white matter is encountered.

This boundary is detected as follows. As in hysteresis thresholding, we use both a lower and an upper threshold. Voxels with grey values above the upper threshold or below the lower threshold are immediately deleted.

If the grey value is between the upper and the lower threshold, we check its immediate neighbourhood which we define to be a window N_n of size $n \times n \times n$, where usually $n = 5$. We compute the grey value distribution of all voxels within $N_{n \times n \times n}$ that have already been deleted in previous iterations or that are below the lower threshold. As these voxels can be assumed to be cortex voxels, we obtain a *local* estimate of the cortical grey value distribution at that point.

The voxel is classified as a white matter point if its grey value is significantly higher than the average grey value of its grey matter neighbours. The deviation is considered significant if it deviates by more than the standard deviation from the mean value.

The segmentation algorithm depends upon two parameters, namely the lower and upper threshold. Good choices for both parameters can be automatically estimated from the image histogram so that the entire segmentation procedure can be performed without manual intervention. So far, we have applied the algorithm to more than 250 high-resolution MDEFT data sets. The accuracy of the results was validated by visual inspection. The algorithm is used only for visualization purposes so that a more exact evaluation of the segmentation accuracy is not necessary here.

3.4.2 Sulcal cut segmentation

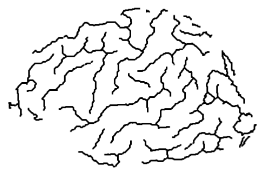


Figure 5. Example of a sulcal cut (left hemisphere).

Sulcal cuts are three-dimensional polygonal lines that represent cortical folds. They are automatically extracted from white matter segmentations of 3D high-resolution structural MR images. Figure 5 shows an example. Sulcal cuts can be rotated in a graphics display in real-time and geometrically linked via mouse-clicks to the 3D reference image. In Lipsia, sulcal cuts are used for visualization purposes. Activation areas can be displayed in conjunction with sulcal cuts. As a sulcal cut rendering is completely transparent, even activations that are deeply buried below the surface can be viewed in spatial relation to the cortical folding of an individual brain.

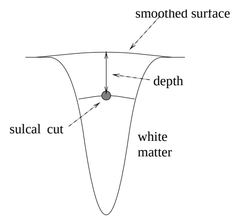


Figure 6. The concept of sulcal cuts. Sulcal cuts are located at a user-defined depth-level below a morphologically smoothed white matter surface.

We define sulcal cuts to be line representations of sulci extracted at one specific level of depth ("cut" at that level). Here, "depth" is defined as the distance towards an idealized smoothed white matter surface. The smoothing is achieved by a morphological closing filter [21]. Figure 6 illustrates the concept of sulcal cuts. For a detailed description of the segmentation algorithm see [22].

3.4.3 Volume rendering

The results obtained by the statistical inference procedures can be visualized using volume rendering techniques. LIPSIA offers two modes of volume rendering that we call 'grey matter rendering', and 'white matter rendering' for lack of a better word.

In grey matter rendering, activation areas are mapped onto a 3D reference image and volume-rendered. Note that activation areas are not geometrically altered in any way. In particular, they are not projected onto the cortical surface. Clipping planes can be specified so that areas below the surface become visible. Figure 10a shows an example.

White matter rendering presupposes that a white matter segmentation has been performed using the high-resolution structural data set. White matter segmented images allow a better view into sulcal valleys and are therefore sometimes preferable. There are two methods of white matter rendering in LIPSIA that are described in the following.

In the first white matter rendering technique, activation areas are displayed together with the white matter segmented image. As in grey matter rendering, activation areas are not geometrically altered in any way. To allow a better perception of sulcal depth, the rendering is augmented with a depth-related color coding. Deeper areas are shown in dark blue, more shallow areas are displayed in light blue. Sulcal depth is estimated as follows. We begin by a morphological closing of the white matter image using a spherical structuring element of size $12 \times 12mm$. After closing, the sulci are filled to the rim. We then compute depth with respect to the morphologically closed surface using a Euclidean distance transform. At the end of this procedure, each voxel has received a depth label. This label is then encoded by a color code. For more information about morphological filters and distance transforms see [23],[21].

As before, it is also possible to specify clipping planes so that deeper areas become visible. In this mode, the depth color coding is omitted. Figures 10b,c show some examples.

We now describe another method of white matter rendering. In contrast to the rendering methods described so far, activation areas are now geometrically transformed and projected onto a surface. This projection surface is defined as follows (see figure 7).

We begin by computing the morphological closure of the white matter segmentation so that the sulci are filled to the rim. On gyral crowns, the projection surface equals the closure surface. Over sulcal valleys, the projection surface is lowered by a user-defined amount. Algorithmically, this lowering is achieved by thresholding the distance transform. Pixels that belong to activation areas are then projected onto the nearest point on the projection surface. If more than one pixel projects to the same point on the surface only the pixel with the largest activation value is displayed.

Only pixels that are close enough to the projection surface are displayed. The maximal distance is user-defined. A typical value is 10mm. Figure 10d shows an example.

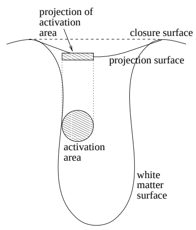


Figure 7. The projection surface used in white matter rendering.

The advantage of this projection method is that it does not distort spatial adjacency. Activated pixels that are adjacent stay adjacent after projection. This is not always the case with surface flattening projections where activation areas may be split up into two parts that are projected onto far apart regions.

Therefore, localization is generally more reliable in our method. On the downside, the size of the activation areas is not preserved.

4 Software design

The LIPSIA software is implemented in C and C++ and runs within the Unix operating system. Programs are started either from the command line, from a Unix-shell-script or from a mouse-driven graphical user-interface. Most programs additionally run in a parallelized version implemented on a 16-processor parallel computer (SGI origin 2000).

The data format used in LIPSIA is based on a software package called *Vista* [24]. Vista offers a versatile data format that can be customized to specific applications. In LIPSIA, a functional MRI time series is stored in one Vista file per subject. The 2D structural slices that are acquired along with the functional data are stored in the same file so that an experiment involving n subjects generates no more than n data files.

Computations times are generally short. For example, a statistical evaluation of an fMRI time sequence consisting of 8 slices with a 64×64 image matrix and 256 scans (timesteps) takes about 7 seconds on an SGI origin 2000 using a single processor. A time sequence consisting of 2280 timesteps with 5 slices and a 64×64 image matrix takes about 6 minutes of computation time when using five SGI origin 2000 processors in parallel. Comparisons with the SPM99-package [25] have shown that these computation times are faster by a factor of 11 and 7, respectively. At the same time, LIPSIA uses less main memory (356 MByte vs. 575 MByte) and fewer files on disk (4 files (107 MByte) vs. 4589 files (368 MByte)) where the numbers refer to the second experiment involving 2280 timesteps.

Figure 8 illustrates the data flow used to process fMRI data. Note that two different processing chains can be used: the registration procedure can be either applied to the preprocessed functional data or it can be applied to the result of a statistical evaluation. The non-linear normalization step can be optionally omitted. Information about the exact processing chain that was used to evaluate a data set is recorded in the image header so that experimental results are reproducible later on.

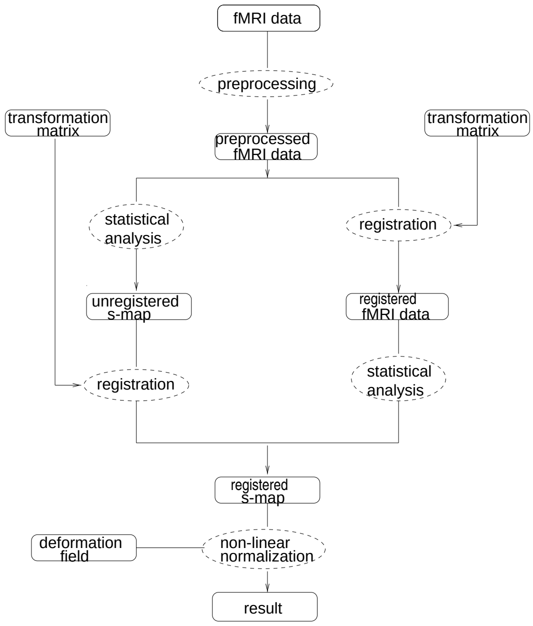


Figure 8. The fMRI data flow diagram. Two different processing chains are possible in which the order of registration and statistical evaluation is reversed. The term 's-map' denotes both z-maps and contrast maps. The transformation matrix and the deformation field are obtained by spatial registration and normalization (see figures 2 and 3).

5 Discussion

We have introduced the software package LIPSIA that is used to process fMRI data. In view of the large amounts of fMRI data, one of the most important aspects in developing LIPSIA was computational efficiency and ease of data handling. This was achieved by using a versatile data format and efficient and in part also parallel implementations.

Various other software packages for the analysis of fMRI data exist, e.g. [26, 27, 28, 25]. We have deliberately used well established statistical algorithms whenever possible. The most widely accepted statistical methodology for evaluating fMRI time series to date is probably the one that is currently implemented in the SPM-package [25]. Therefore, we have adopted the same statistical and hemodynamic modeling approach. Direct comparisons of both packages show that when the LIPSIA-preprocessing, registrations and normalizations are used, the results of the statistical evaluations are almost exactly alike in both packages. However, due to a more efficient software implementation, computation times in LIPSIA are faster by a factor of about 7 or more. Depending on the experimental parameters this factor may often even be higher. Also, the data format used in LIPSIA is easier to handle.

Other parts of LIPSIA are quite different from other comparable software packages. This is the case with the registration and normalization routines, as well as the visualization and segmentation package. Our main objective here was to support the analysis of individual brain anatomy. Several alternative visualization and segmentation techniques have been introduced with this purpose in mind.

References

[1] J.W. Belliveau, D.N. Kennedy, R.C. McKinstry, B.R. Buchbinder, R.M. Weisskopf, M.S. Cohen, J.M. Vevea, T.J. Brady, B.R. Rosen. Functional mapping of the human visual cortex by magnetic resonance imaging. *Science*, 254:716–719, 1991.

[2] Talairach J, Tournoux P. *Co-Planar Stereotaxis Atlas of the Human Brain*. New York: Thieme, 1988.

[3] F. Kruggel, G.Lohmann. Automatical adaptation of the stereotactical coordinate system in brain MRI data sets. In J. Duncan, editor, *Int. Conf. on Information Processing in Medical Imaging (IPMI 97)*, Poultney, Vermont, USA, June 9–13 1997.

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition edition, 1992.

[5] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[6] K. J., Fletcher, P., Josephs, O., Holmes, A., Rugg, M. D., Turner, R. Friston. Event-related fMRI: characterizing differential responses. *NeuroImage*, 7:30–40, 1998.

[7] K.J., Frith, C. D., Turner, R., Frackowiak, R. S. J. Friston. Characterizing evoked hemodynamics with fMRI. *NeuroImage*, 2:157–165, 1995.

[8] O., Turner, R., Friston, K. J. Josephs. Event-related fMRI. *Human Brain Mapping*, 5:243–248, 1997.

[9] K. J., Frith, C. D., Frackowiak, R. S. J., Turner, R. Friston. Characterizing dynamic brain responses with fMRI: a multivariate approach. *NeuroImage*, 2:166–172, 1995.

[10] G. Lohmann, K. Müller, V. Bosch,H. Mentzel,S. Hessler, L. Chen, D.Y. v. Cramon. Lipsia - leipzig image processing and statistical inference algorithms. Technical report, Max-Planck-Institute of Cognitive Neuroscience, Leipzig, 2000.

[11] B. J. Winer, D. R. Brown, and K. M. Michels. *Statistical principles in experimental design*. McGraw-Hill, New York, 1991.

[12] Seber, G.A.F. *Linear Regression Analysis*. Wiley, New York, 1977.

[13] K. J. Worsley and K. J. Friston. Analysis of fMRI time-series revisited – again. *NeuroImage*, 2:359–365, 1995.

[14] V. Bosch. Statistical analysis of multi-subject fMRI data: Assessment of focal activations. *J. Magn. Reson. Imaging*, 11:61–64, 2000.

[15] J.C. Bezdek, L.O. Hall, L.P. Clarke. Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4):1033 – 1048, 1993.

[16] L.P. Clarke, R.P. Velthuizen, M.A.Camacho, J.J. Heine, M. Vaidyanathan, L.O. Hall, R.W. Thatcher, M.L. Silbiger. Mr segmentation: Methods and applications. *Magnetic Resonance Imaging*, 13(3):343 – 368, 1995.

[17] R. Guillemaud, M. Brady. Estimating the bias field of MR images. *IEEE Transactions on Medical Imaging*, 16(3):238–251, June 1997.

[18] W.M. Wells, W.E.L. Grimson, R. Kikinis, F.A. Jolesz. Adaptive segmentation of MRI data. *IEEE Transactions on Medical Imaging*, 15(4):429–442, Aug. 1996.

[19] G. Borgefors. Distance transforms in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[20] Y.F. Tsao, K.S. Fu. A parallel thinning algorithm for 3d pictures. *Computer Graphics Image Proc.*, 17:315–331, 1981.

[21] J. Serra. *Image Analysis and Mathematical Marphology.* Academic Press, 1982.

[22] G. Lohmann. Extracting line representations of sulcal and gyral patterns in MR images of the human brain. *IEEE Transactions on Medical Imaging*, 17(6):1040–1048, December 1998.

[23] G. Lohmann. *Volumetric Image Analysis.* John Wiley & Sons, Chichester, England, 1998.

[24] A. Pope, D. Lowe. Vista: A software environment for computer vision research. In *Int. Conf. on Computer Vision and Pattern Recognition*, Seattle, Wash., 1994. (see also: `http://www.cs.ubc.ca/labs/lci/vista/vista.html`).

[25] Wellcome Department of Cognitive Neurology. *SPM99.* London,England. http://www.fil.ion.ucl.ac.uk/spm/.

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

[27] R. Goebel. BrainVoyager 2.0: from 2D to 3D fMRI analysis and visualization. *NeuroImage*, 5:S635, 1997.

[28] J.P. Strupp. Stimulate: A GUI based fMRI analysis software package. *NeuroImage*, 3:S607, 1996.

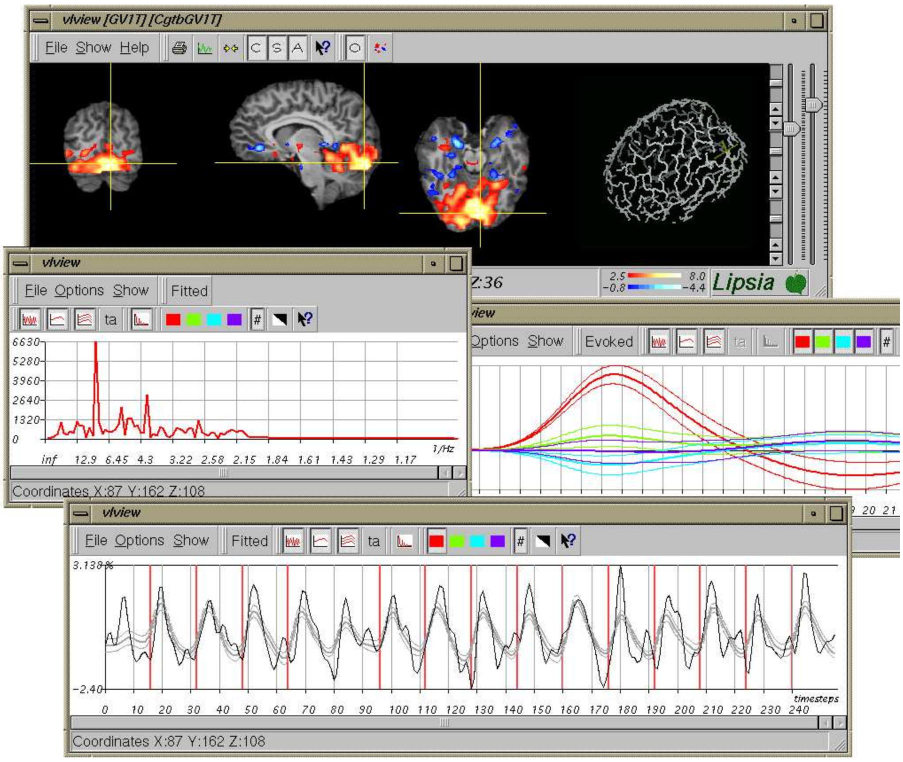
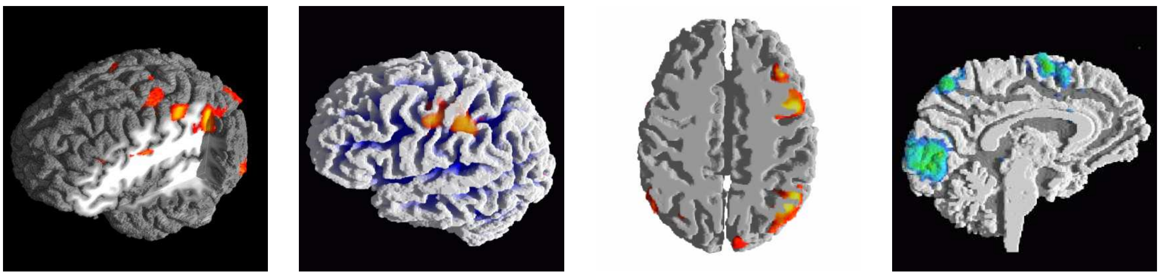


Figure 9. Displaying cross-sectional slices, sulcal cuts and time courses. The subwindows show a time course together with experimental design information, a power spectrum, and the result of a statistical modeling obtained from several event types.



a) b) c) d)

Figure 10. a) Volume rendering using clipping planes; b) depth-color coded white matter rendering; c) white matter rendering with clipping plane; d) projection of activation areas onto a morphologically closed cortical surface (left median surface).