Institut National Polytechnique de Grenoble
Laboratoire des Images et des Signaux
46, avenue Félix Viallet
38031 Grenoble cedex

Rapport de Master

Daniel Cho
Master Sciences Cognitives 2003-2004

Evaluation des méthodes de séparation de sources: ICA appliquée à l'IRM fonctionnelle

Date de soutenance: 21 juin 2004

Responsables de stage :
Christian Jutten, professeur des universités, INPG
Michel Dojat, ingénieur de recherche, INSERM

Acknowledgements

Special thanks must be given to Monsieur Christian Jutten for instructing and guiding me through the process of this stage and understanding more deeply this very interesting subject. I have always appreciated his extensive knowledge in this area, his valuable insights, time and patience to answer any of my undoubtedly confused questions.

I would also like to thank Monsieur Michel Dojot at INSERM for providing the actual data from his experiments contained in this paper and all his valuable assistance in helping me to understand more fully fMRI experiments.

Contents

Chapter 1

Introduction

The purpose of this paper is to evaluate one of the most recently developed methods to the problem of separation of source, Independent Component Analysis (ICA), and to apply this method concretely to the analysis of fMRI data.

Independent Component Analysis addresses the problem of trying to discover the underlying factors of multivariate data without knowing their mixing nature a priori. ICA has now emerged as one of the prime techniques to this problem known as blind source separation (BSS). Though many methods exist for source separation, the key distinguishing trait of ICA is that it solves the problem by giving solutions that are statistically independent and nongaussian.

ICA has attracted high interest in recent years from numerous different research communities including machine learning, neural networks, statistical signal processing, and Bayesian modeling. Indeed from a machine learning standpoint, ICA can be viewed as belonging to the area of unsupervised learning since the underlying components must be learned from the data without any supervising input from a teacher. Though it can be grouped next to these other feature extraction techniques, ICA is elicits quite a different paradigm from traditional machine learning methods by introducing the diametric concepts of data-driven versus hypothesis-driven methodology.

When as little as possible is known about some data set, the central problem of analyzing multivariate data has been to find out if an underlying structure to the seemingly random data exists. The exploration of blind source separation addresses the question of how to decompose observed signals when one does not know the nature of the sources nor their mixing matrix. This problem of separation of source has historically been approached from a lower-order statistical framework, notably factor analysis and subsequently principal component analysis (PCA). One can see the progression in research from mutual information to PCA to ICA. However ICA is fundamentally different from these other techniques in that it uses higher order statistics and makes a few key assumptions: the system is composed of a linear combination of statistically independent variables, the distribution of which is nongaussian.

ICA has been applied to a diverse range of problems. Applications of ICA have included adaptive speech filtering, speech analysis, signal coding, biomedical signal processing, image compression, text modeling, and even financial time-series analysis. The problem of separating mixed audio signals picked up from different sensors is the classic example that ICA is able to handle very well.

Taking another example in financial time-series data, the fluctuations of equity prices can be considered as the resulting signal of a myriad array of input factors such as interest rate levels, inflation, unemployment, political stability, even optimistic or pessimistic mindset of the consumer. While economic data such as inflation are well quantified and recorded, other factors such as political stability and people's mindset to the markets are less easily quantifiable. ICA provides a powerful solution to this by finding the underlying factors, as well as their "weights" in terms of their mixing coefficients, that are driving the end result for equity prices. In this regard, ICA is known for its dimension-reducing ability, and as such it has been used as a pre-processing tool to transform a time-series into a lower-dimensional data set [4].

In the domain of biomedical signal processing ICA has been actively used to study EEG and fMRI data. This paper will concentrate on the application of ICA to fMRI experiments. In many respects fMRI is an ideal target for ICA application since the registered information of cerebral activity can be regarded as a linear mixture of source signals; in this case being neurophysiological functions such as vision activity, language processing, and physical movement. Brain activity in any particular region of the brain could be the result of any combination of these artifacts, thus the mixture of signals is unknown. In this study ICA is applied to fMRI experiment data involving vision-related tasks, as both temporal and spatial structure are computed.

Chapter 2

Independent Component Analysis

2.1 Introduction

Independent Component Analysis seeks to distinguish the underlying factors giving rise to a set of observation signals. When the sources are unknown as well as their mixing matrix the problem of discovery of the sources is known as blind source separation (BSS).

The often cited situation of the cocktail mixing party is a good springbroard to develop the idea of BSS problems and the application of ICA. Imagine one is at a conference where two people are overheard talking simultaneously. The meeting is recorded by two microphones placed in different locations. If the recorded time signals are denoted by $x_1(t)$ and $x_2(t)$, each signal can be regarded as the weighted sum of the source signals originating from the speakers, $s_1(t)$ and $s_2(t)$. Then this relationship can be expressed by the linear equations:

$$x_1(t) = a_{11}\, s_1 + a_{12}\, s_2$$

$$x_2(t) = a_{21}\, s_1 + a_{22}\, s_2$$

This can be extended to n number of source signals and m number of recording instruments:

$$x_i(t) = a_{m1}\, s_1 + a_{m2}\, s_2 + \ldots\ldots\ldots + a_{mn}\, s_n$$

Thus in general an unknown mixed signal can be represented as:

$$x_i(t) = \sum_{j=1}^{N} a_{ij} s_j(t), \;\; j = 1, \ldots, N$$

where N observed signals x_1(t), ..., x_N (t) are linear combinations of N statistically independent source signals $s_1(t)$, ..., s_N (t), and a_{ij} are the elements of the unknown mixing matrix. This gives us the linear problem:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

where the square matrix ***A*** is known as the mixing matrix. By assuming it is invertible, the goal is to find an *N* x *N* separating matrix such that the output vector ***y*** = ***Bx*** is an estimate of the source vector ***s***.

$$\hat{\boldsymbol{s}} \equiv \boldsymbol{y} = \boldsymbol{Wx}$$

ICA attempts to solve this equation by finding the square matrix ***W*** such that the components of the linearly transformed vector ***y*** are mutually statistically independent.

In order to find ***W***, several assumptions are made. The sources are statistically independent and give rise to linear combinations of the observed signals. This is a generally acceptable assumption with regards to fMRI analysis since localized brain activity is considered as the generation of linear combinations of various physiological effects. In addition the sources are expected to be non-gaussian and stationary.

It should be noted that the observations do not have to be the result of the mixing of independent sources. However it may still be of interest in finding the representative ***y*** vector. In seeking components that are as statistically independent as possible, ***y*** would signify the inherent underlying structure to the system. In this way problems can be simplified for interpretation, and is in fact often used as a pre-processing stage to reduce dimensionality.

In order to find the linear transformation from the observation vector ***x*** to ***y***, ICA seeks to maximize joint entropy and minimize mutual information. Mutual dependence is usually measured by:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{y})$$

where *H* denotes Shannon's entropy and *I*(***y***) signifies the amount of information that is shared by the components of ***y***. The mutual information of the components is zero only when they are mutually independent. For two variables the definition of independence is:

$$p(\mathbf{s}) = p(\mathrm{s}_1)\, p(\mathrm{s}_2)$$

Thus for a multivariate system, mutual independence occurs when their joint probability density is equal to the product of the marginal densities:

$$p_y(\mathbf{u}) = \prod_{i=1}^{n} p_{y_i}(u_i)$$

A close connection connection between minimizing mutual information and maximizing likelihood can be seen by the Kullback-Leibler Information Number:

$$I_{K-L}(g:f) = \int g(y) \log\left\{\frac{g(y)}{f(y)}\right\} dy$$

This formula measures the degree of distance between probability measures, i.e., the similarity between the statistical model and the true distribution. As *I*(***y***) is equivalent to the Kullback-Leibler divergence between the products of the marginal densities and the actual joint density p(***y***), it proves to be good measure of the dependence of components.

2.2 Difference between ICA and PCA

For heuristic as well as procedural purposes Principal Component Analysis (PCA) can be a valuable aid to understanding and better appreciating the unique traits of ICA.

ICA can in some respects be seen as a non-gaussian version of factor analysis and PCA. Indeed the seminal work into BSS by Jutten and Herault (1991) [3] looked at extensions to standard PCA, outlining the transition from PCA to ICA. While both methods decompose signals into components, the key difference is the statistical type of components obtained. The goal of PCA is to find components that are just uncorrelated, a second-order statistical property. It gives the projections of the data in the direction of maximum variance. Thus each principal component are ordered according to their variance. The first eigenvector component gives the direction with maximum possible variance, the second gives the subsequent maximum variance in the remaining subspace, etc. In this way PCA does not minimize the mutual information among transformed components, but merely decorrelates them. ICA on the other hand takes into account higher-order dependencies whereas PCA is based only on second-order covariance matrices.

Independence is stronger than uncorrelatedness: independence implies nonlinear uncorrelatedness. For example if s_1, s_2 are independent, then any nonlinear transformations g(s_1), g(s_1) are uncorrelated (covariance = 0). If two random variables are merely uncorrelated, nonlinear transformation do not have generally have zero covariance.

It has been shown that one major difference is that spatial components in PCA are assumed to be orthogonal, whereas ICA spatial maps are not. Other assumptions made in PCA become the opposite constraints in ICA. While random variables in PCA are assumed to be gaussian, independent components in ICA are required to be nongaussian. For if one assumes that the mixing matrix is orthogonal and the source signals are gaussian, then their joint density given by:

$$p(x_1, x_2) = \frac{1}{2\pi}\exp(-\frac{x_1^2 + x_2^2}{2})$$

would produce a density distribution totally symmetric. In no particular direction is any information contained in the directions of the columns of the mixing matrix **A**, thus preventing it be estimated.

This is an important point also for if the components were actually gaussian PCA would always produce independent components. But this would happen only in exceptional cases, showing that most data are nongaussian. Rather than being a restriction, this requirement is a more realistic assumption. Thus with these significant differences ICA is able to find underlying factors when other methods such as PCA fails.

ICA can be viewed as a post-processing method for oblique rotation of PCA data. In this regards PCA is applied as a first step in a pre-whitening stage; that is the components are de-correlated and their variances are made to equal unity. Thus the covariance matrix equals the identity matrix:

$$E\left\{\tilde{x}\,\tilde{x}^{T}\right\} = I$$

Using the eigenvalue decomposition of the covariance matrix:

$$E\left\{\tilde{x}\,\tilde{x}^{T}\right\} = EDE^{T}$$

where E on the right side of the equation is the orthogonal matrix of eignenvectors on the left side. D is a the diagonal matrix of its eigenvalues, $D = diag(d_1, \ldots ,d_n)$. The whitening is achieved by:

$$\tilde{x} = ED^{\frac{-1}{2}}E^{T}x$$

Substituting for ***x***, one obtains a newly transformed mixing matrix:

$$\tilde{x} = ED^{\frac{-1}{2}}E^{T}As = \tilde{A}s$$

The advantage of Ã is that the number of parameters to be estimated are reduced as it is now orthogonal:

$$E\left\{\tilde{x}\,\tilde{x}^{T}\right\} = \tilde{A}E\left\{\tilde{s}\,\tilde{s}^{T}\right\}\tilde{A}^{T} = \tilde{A}\tilde{A}^{T} = I$$

Since an orthogonal matrix contains only about half of the number of parameters of general matrices, whitening helps to reduce the complexity of the problem.

2.3 State of the Art and Latest Algorithm Implementations

In this section several state of the art extensions and implementations of ICA will be briefly reviewed: Infomax, MISEP, PICA, and FastICA. In this study of fMRI data however the FastICA algorithm was used as the focus of calculations.

Infomax and MISEP

Derived from a neural network viewpoint, Infomax is a contrast function that uses the principle of maximum information preservation in order to estimate the independent components. Through the close equivalence to maximum likelihood estimation, Infomax attempts to maximize the output entropy or "information flow" from a a neural network with nonlinear outputs:

$$L = H(g_1(\mathbf{w}_1^T\mathbf{x}),...,g_n(\mathbf{w}_n^T\mathbf{x}))$$

Here $\boldsymbol{x}$ is assumed to be the inputs to the neural network while $g_i(\mathbf{w}_i^T\mathbf{x})$ are the outputs in the form of nonlinear scalar functions and $\boldsymbol{w}_i$ are the weight vectors of the neurons.

Most computations using ICA assume a linear mixture. While lacking a general analytic solution, the problem is at least well-posed. Ill-posed nonlinear mixtures on the other hand have been the focus of much more recent research. MISEP is one such method that is versatile enough to handle both linear and nonlinear ICA based on mutual information. It is basically an extension of Infomax for the handling of nonlinear mixtures as well as for the learning of the nonlinearities at the outputs of the network. Nonlinear blind separation problems are generally ill-posed; however, when the nonlinear mixture is relatively smooth, MISEP is able to solve the problem through the use of regularization.

PICA

Probabilistic ICA (PICA) as implemented in the FSL-Melodic package from Oxford University takes a different tract altogether by using Bayesian statistics to extract independent components. In addition to their stated advantage in overcoming of the problem of overfitting, another chief advantage is the prediction of the number of components. This is calculated using Bayesian dimensionality estimation techniques.

PICA permits non-square mixing in the presence of Gaussian noise. In order to avoid overfitting, objective estimation of the amount of Gaussian noise is employed through Bayesian analysis of the actual dimensionality of the data. That is discounting artifacts and Gaussian noise, only the number of activations and non-Gaussian noise sources are counted. In this way probabilistic modeling is able to be generate an asymptotically unique decomposition of the data. It reduces the problem of interpretation of data by making it much more likely that each final independent component is due to only one physiological process.

Probabilistic ICA is akin to a generative linear latent variables model. It is assumed that the multivariate vector of observations is generated from a set of statistically independent non-Gaussian sources linearly intermixed, corrupted however by an addition of Gaussian noise ***η(t)*** :

$$x_i = As_i + \mu + \eta_i \quad \forall i \in \nu$$

The mean of observations $\boldsymbol{x_i}$ is defined by the vector $\boldsymbol{\mu}$. Like other ICA methods a solution to the equation is sought in isolating ***s*** by finding a linear transformation matrix ***W***. However there are marked differences.

Interestingly the PICA model bears some similarities to the standard General Linear Model (GLM). However, unlike the design matrix in GLM, the mixing matrix is not pre-specified prior to model fitting but is estimated from the data as part of the model fitting. Thus PICA remains a data-driven model. The spatial source signals correspond to estimates in the GLM with the constraint that they be statistically independent.

Another unique feature about PICA is the incorporation of a priori knowledge about the spatiotemporal response of source processes. Spatial and temporal information such as the assumption of spatial smoothness of the areas of activation or temporal autocorrelation is integrated into the model by temporal-spatial filtering of the data prior to model fitting. Other improvements to standard ICA include temporal pre-whitening and variance normalisation of timeseries, the latter being especially useful for reduction of dimensionality when activation is weak. Due to all these extensions specifically targeted to cerebral imaging, PICA appears to be particularly suited to FMRI data.

FastICA is a fixed-point iteration scheme that was developed and maintained at Helsinki University of Technology. Among its advantages are speed as it is generally faster than conventional gradient descent methods. This algorithm will be explained in more depth in the following section.

2.4 FastICA Algorithm

The FastICA algorithm is a fixed-point iteration approach to independent component analysis. The goal as always is to find the direction where the projection of $W^T x$ maximizes its nongaussianity.

According to the central limit theorem, sums of nongaussian random variables are closer to being gaussian than the original variables. Kurtosis, a fourth order statistical property, and negentropy, a derivative of entropy, are both measures of nongaussianity. FastICA uses negentropy as it is a more robust measure of nongaussianity. Based on a modified version of differential entropy, negentropy is defind by:

$$J(y) = H(y_{gauss}) - H(y)$$

where y_{gauss} is a Gaussian variable of the covariance matrix y and H is the entropy of the discrete random variable as defined by :

$$H(y) = -\sum_i P(y = a_i) \log P(y = a_i)$$

where the possible values of y are a_i.

As the actual estimation of negentropy proves difficult, in actual implementation for the pursuit of nongaussian maximization, a few types of Newton iteration can be used:

$$g_1(u) = \tanh(a_1 u)$$
$$g_2(u) = u \exp(-u^2/2) \quad 1 \le a_1 \le 2$$

where a_1 is some suitable constant, often taken as $a_1 = 1$.

A number of distinct traits of FastICA as compared to the other algorithm methods for ICA should be noted.

- Convergence is fast: cubic or at least quadratic. In contrast most other ICA algorithms converge linearly since they are based on stochastic gradient descent methods.
- Unlike most gradient-based methodology there are no step sizes to choose.
- Various functional forms of the g function can be chosen such as power3 (u^3), tanh, gauss, skew (u^2).
- Approach to estimate the independent components can be deflation or symmetric which solves for the components serialy or simultaneously respectively.

Chapter 3

fMRI Experiments

3.1 Introduction

Magnetic resonance imaging has offered new and powerful ways to approaching the study of cognitive neuroscience. Indeed it can be said that in many ways it has revolutioned medicine as well as cognitive studies of the brain. Before delving into MRI a few other cerebral analysis techniques will be briefly reviewed.

PET

In the past before MRI another computer imaging technique known as positron emission tomography (PET) has been used to examine the brain for medical purposes, but this did not lend itself to effective research purposes. PET scans require injection of positron-emitting isotopes, a radioactive substance, in order to distinguish human tissue. This technique is very useful for detecting tumours; however, for clinical research purposes unnecessary radioactive injection would be undesirable. Another impediment is that since the half-lives of the radionuclides are very short, the time lengths for experimentation are effectively limited by the nature of the testing tool. Further the requirement of multiple acquisitions prolong image processing times. While both PET and MRI use computer analysis of atomic detection, they are of an altogether different class. Tomographic images are formed by the computer analysis of the paths of photons (gamma rays) being detected due to the annihilation of positrons colliding with electrons emitted by radionuclides. MRI detect magnetic spin of deoxyhemoglobin.

EEG

Electroencephalography (EEG) has long been and is still a common diagnostic tool (despite the fact that it was discovered before the 20th century) simply by way of its lower cost and accessibility. Electrodes are attached to the scalp, and electrical activity of the brain are measured, resulting in graphs of so-called brain waves. The recording of synaptic activity is enabled by the potentials generated by the nerve cells during some functional task or other neurological activity. The dominant frequency of these potentials can range from 8 to 10 cycles per second and the amplitude about 10 to 100 microvolts. Thus highly fine temporal resolution can be obtained through EEG. In functional experiments fluctuations in potential can be correlated with neurologic tasks. This mapping technique is equally used in fMRI experiments and was done so in this study. As stated its advantages are low cost, accessibility, and non-intrusive methodology. Another powerful asset is the lack of time restraints. There even exist overnight EEG devices designed to check the electrical activity in brains of a sleep-deprived patients for example. In addition, 48-hour mobile units are used to increase the chance of capturing seizure activity of epileptic patients. Its chief advantage over fMRI is its higher temporal resolution, however it fares poorly in the area of fine localization and spatial resolution.

EEG data can be hard to analyse as the levels of noise can be high; for example, muscle movement such as eye blinking or other artifacts blur the targeted signals of interest. Traditional methods have removed the noise by taking an average of signals, but this renders the detection of local events, such as spikes, much less accurate. ICA has thus been used to isolate the independent components of interest.

MRI

MRI has its roots in the principles of nuclear magnetic resonance (NMR), a spectroscopic technique used to obtain microscopic chemical and physical information about molecules. MRI started out as a tomographic imaging technique by producing an image of the NMR signal as a thin slice through the human body. MRI has well advanced beyond tomographic imaging to a volume imaging technique. In fact among its chief strengths is its capacity for very high resolution along with its non-invasive nature.

Nuclear particles have a spin characteristic that behave much like magnets within a magnetic field. MRI takes advantage of this fundamental spin. A large magnet is applied to polarize hydrogen atoms in the tissues and then monitors the summation of the spinning energies within living cells.

Functional MRI is based on the increase and decrease of blood flow to the brain that occurs during neuronal activity. Thus mental tasks cannot be observed directly but indirectly by haemodynamic responses caused by those tasks. It is actually the resulting local reduction of deoxyhemoglobin that is measured and detected as activity in that region. This due to the fact that increase in blood flow occurs without a corresponding increase in oxygen extraction. Since deoxyhemoglobin is parametric due to its high composition of iron, it alters the T2* weighted magnetic resonance image signal. The observed T2* is dependent on the presence of blood deoxygenation. The deoxygenated haemoglobin in turn is "blood oxygenation level dependent" (BOLD) and the BOLD effect that can be observed by in high magnetic fields generated by MRI instruments.

In functional MRI experiments, resolution is lowered and images taken in rapid succession concurrent with some functional task. A detailed three-dimensional image of the highest resolution is usually taken first and then during the run of the experiment lower resolution images are taken in order for faster processing. Below in Figures 1 and 2 are the high resolution anatomical images followed by a set of the images at lower resolution captured during the experiment taken at t=5. fMRI temporal resolution ranges from 1 to 2 seconds and spatial resolution ranges from 1 to 3 mm^3. A complete detailed anatomical image of the brain has a resolution of 157x189x136 voxels where each voxel is 1x1x1 mm^3. The functional lower resolution images have a resolution of 40x48x28 voxels, where each voxel is 4x4x5 mm^3.

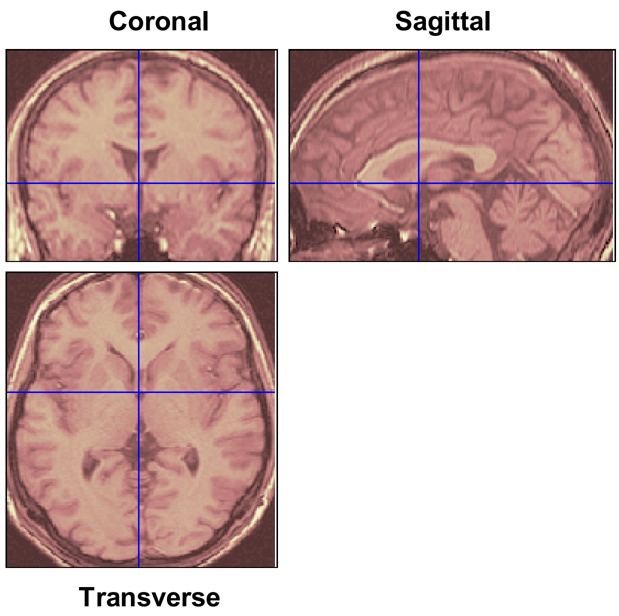


Fig 1. *A detailed, normalised brain image of the subject taken by a statistical parametric map (SPM). Resolution of voxels is at 1x1x1 mm, while the whole image contains a voxel matrix 157x189x136, much finer than during normal functional image processing during experiment. The images are, clockwise, coronal, sagittal, and transverse.*

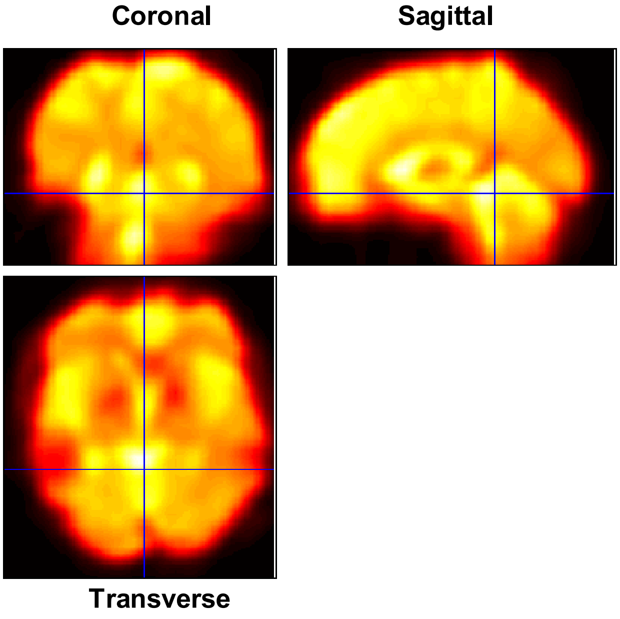


Fig 2. *Funtional images taken at t=5 (total 144). Notice the resolution is lower to allow for faster processing, during a functional MRI experiment. Voxels now are at 4x4x5 mm and the total image contains a matrix of 40x48x29 voxels. Crosshairs are at voxel [21 19 10] (x=0, y=-40, z=10).*

3.2 Block Vs. Event

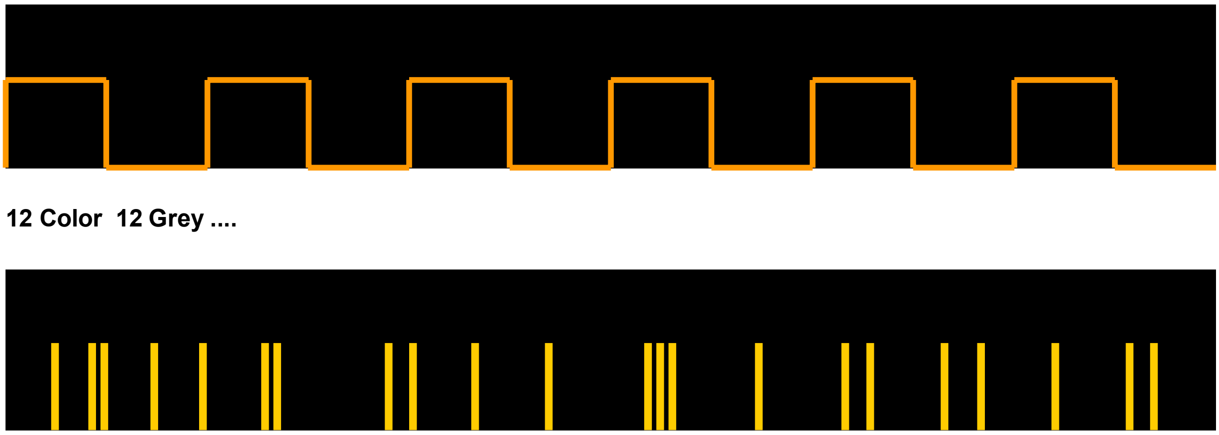


Fig 3. Above is a representation of a block-designed experiment run. Below is a model of an event-related design.

In functional magnetic resonance imaging, experiments are usually carried out in one of two different fashions: blocked designs and event-related designs. Both have their advantages and disadvantages. Blocked design experiments consist of the presentation of stimuli in blocks of fixed-time periods followed by equal time periods of repose. Thus the effects of interest are obtained by a type of boxcar regressor convolved with a hemodynamic response function. The evident assumptions for this type of model are that steady-state synaptic activity and hemodynamics are reached within each block time-period.

Event-related designs, on the other hand, intersperse stimuli in rapid and random fashion among rest periods. Here the design is more akin to the impulse response of hemodynamic activity to quick stimuli. Experiments can also be designed with a combination of the two approaches.

In this particular experiment the block-design consisted of a block of 12 different color images that were presented in succession followed by a block of 12 grey images for a total of 6 blocks each. In the event-related design images of color, grey and null were presented in a random order for very brief periods of time.

A major limitation of blocked design experiments is that they allow the observers to anticipate, generate response strategies, or allocate their attention differently across blocks. Moreover, blocked designs are time consuming, somewhat boring and tiring for the subjects. These factors may modulate differentially the fMRI activations observed across conditions.

Event-related designs avoid these problems by presenting stimuli in a random fashion where the observer cannot anticipate when and what is being shown and allows for the adaptation of behavioral paradigms to fMRI experiments. Numerous human fMRI studies have used this method for testing neural responses to various cognitive tasks (e.g. oddball detection, selective attention, working memory tasks). However there is one large drawback to this in terms of the application of ICA. Due to the noise in the signal and very short transitions between activity events a time-course for an event-related series would not produce distinguishable results. Even in block designs one must be wary of the adaptative properties of neurons. Adaption is a dynamic property of neural populations that is indicated by decreased response to stimuli that have been presented repeatedly or for prolonged time periods. Thus it is possible that towards the latter stages of an experiment decreased activity range may be detected.

3.3 Imaging Tools

Two different imaging tools were used in this study, SPM and FSL. SPM was developed at Wellcome Department of Imaging Neuroscience of University College London. FSL was developed at the Oxford Centre for Functional Magnetic Resonance Imaging of the Brain of Oxford University. While SPM runs on both Windows and Linux platforms, FSL is best used on Linux and other UNIX derivative platforms. The latter can be run under Windows using emulation software but stability issues are not certain. In terms of image viewing both software packages provide similar functionality. However FSL includes a module known as Melodic which does ICA analysis using the PICA algorithm that was discussed in an earlier section.

3.4 SPM

Statistical Parametric Mapping (SPM) is an implementation of the General Linear Model (GLM) for the analysis of fMRI and PET scan data by way of a spatially extended statistical process. The software package is built on Matlab (with some C routines) and SPM image files use the Analyze file format that is common to brain imaging software (see Appendix B for screenprints)

In SPM images are spatially normalised and voxels smoothed by convolving the individual volumes with a Gaussian smoothing kernel. The vast majority of analytical techniques currently applied to FMRI data test specific hypotheses about the expected BOLD response using simple regression or more sophisticated models like the General Linear Model (GLM). The expected signal changes are specified as regressors of interest, and the estimated regression coefficients are tested against a null hypothesis. These voxel statistics generate the images known as statistical parametric maps.

This approach is confirmatory in nature and make strong a prior assumptions about the spatio-temporal nature of the signals. This type of hypothesis-driven approach differs significantly from data-driven methods such as ICA. One of the problems that could arise with hypothesis-based techniques such as this is the possible presence of unmodeled artifactual signal in the data. N noise which is temporally non-orthogonal to an assumed regression model will bias the parameter estimates, while noise orthogonal to the model will inflate the residual error. In both cases any divergence between the assumed and real signal signal space will reduce the statistical accuracy of SPM.

3.5 Experiment Procedure

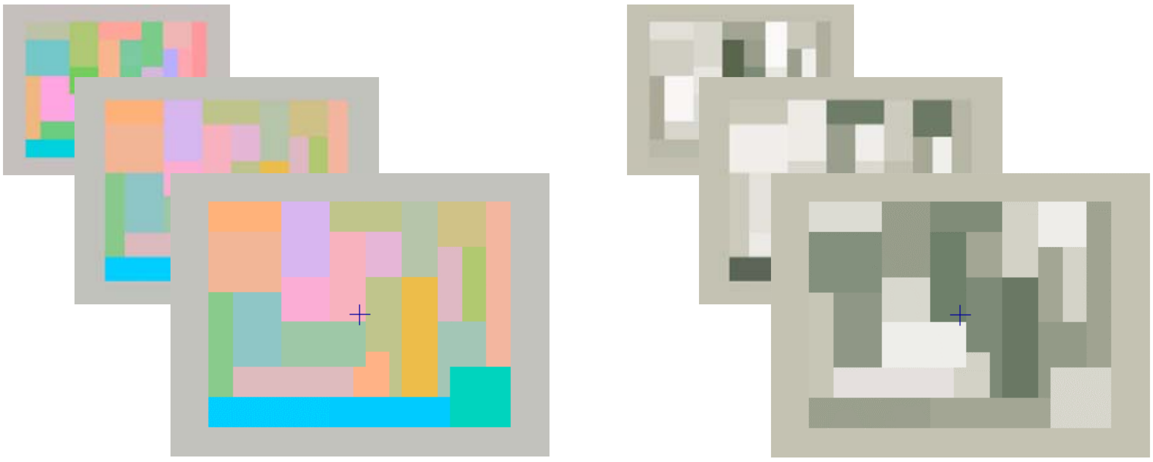


Fig 4. *3 samples out of the 12 color images and 12 grey images that were presented to subject.*

The focus of this experiment was on visual tasks[1]. Three separate runs under the blocked design and four for the event-related design were executed in this experiment. For the blocked design 12 different color images were shown for 3 seconds each for a total of 36 seconds within one block. This was followed by an equal time period of 12 grey images, providing the contrasting activity to counter the BOLD excitation. Thus a total of 6 alternating blocks of stimuli and rest comprised a total of 144 images. Samples of the color stimuli and grey images are shown above in Figure 4.

For the event-related design color, grey and rest periods lasting 500 ms were flashed in a random fashion to the subject. The order of the types of image is listed in the tables below for the first run and second and third runs respectively

Table 1 – First Run

Color
0 2 3 4 5 11 19 22 43 44 46 47 48 49 54 56 58 62 64 66 73 74 75 76 77 79 80 81 82 83 84 88 90
Grey
6 7 8 9 10 15 16 17 18 25 26 29 31 33 34 35 36 37 40 41 42 45 52 53 55 57 59 61 69 70 71 85 89
Null
1 12 13 14 20 21 23 24 27 28 30 32 38 39 50 51 60 63 65 67 68 72 78 86 87 91 92 93 94 95 96 97 98

Table 2 – Second and Third Runs

Color
1 2 9 14 16 17 18 24 25 31 32 34 39 44 45 47 50 59 66 67 68 69 71 72 73 74 76 77 83 85 86 87 88
Grey
3 4 5 6 7 11 12 19 20 26 27 29 36 37 38 40 43 48 51 52 53 57 60 61 62 63 75 78 79 80 82 84 89
Null
0 8 10 13 15 21 22 23 28 30 33 35 41 42 46 49 54 55 56 58 64 65 70 81 90 91 92 93 94 95 96 97 98

[1] All experiment data was provided by Michel Dojat of INSERM

Chapter 4

Independent Component Analysis of fMRI Data

4.1 Introduction

The FastICA algorithm was applied to the fMRI data from the experiments, and appropriate activation maps and associated time-courses were obtained. Since this experiment focused on visual tasks temporal resolution concentrated on spatial IC's resolved in the visual cortex area. Thus specific planes pertinent to the visual cortex were taken, z = 8, 12, 14.

For the spatial component analysis, the data matrix $\boldsymbol{x}_S$ consisted of 144 rows of observations and 1920 columns representing the voxels per volume (40 x 48). The matrix contained 99 row observations for the event-based run. Thus each row was a vectorization of the entire plane at each instance in time.

$$\boldsymbol{x_s} = \mathbf{A}_S\boldsymbol{s_s}$$

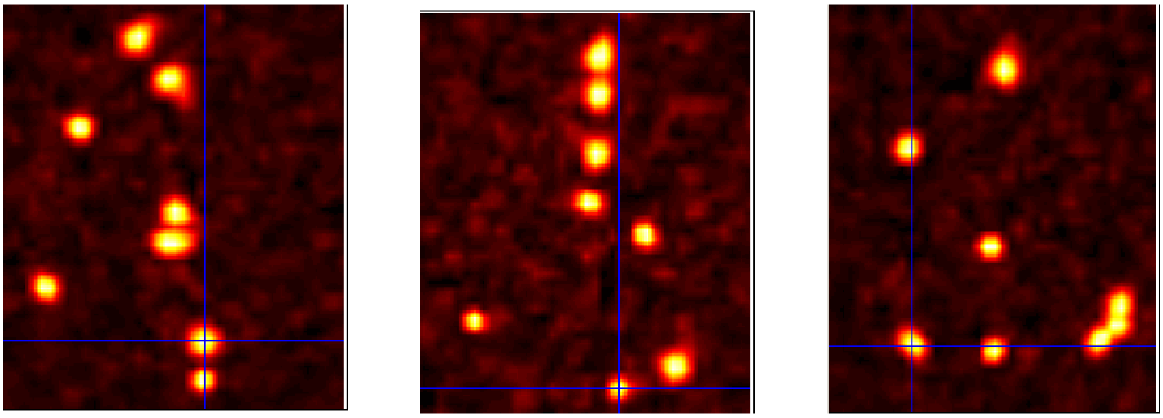
For the time-course computation a small selected portion of the volume was chosen, concentrating on one of the spatial IC's in the visual cortex. Thus only 15 voxels consisting of one active spatial region given by the spatial analysis was extracted. After reducing the columns of $\boldsymbol{x_s}$, the transpose was taken to create the new matrix $\boldsymbol{x_t}$ to calculate the time-coures.

$$\boldsymbol{x_t} = \boldsymbol{x^T{}_s}$$
$$\boldsymbol{x_t} = \boldsymbol{A_t s_t}$$

By taking the transpose and reducing the spatial vectors the temporal components are priveleged. Even still the concern was that there was not enough temporal points and thus the algorithm was likely to overlearn.

In addition specific parameters had to be chosen as arguments to the algorithm. The number of components estimated was 8 for spatial decomposition and 6 for temporal decomposition. The nonlinear g function chosen was power-3 and approach was deflation.

4.2 Spatial Components – Block Design



z = 14 (+15 mm)

z = 12 (5 mm)

z = 8 (-15 mm)

Fig 5. *8 spatial components from the block-designed run extracted by FastICA algorithm on three different z planes visualized by SPM.*

The above images represent the calculations of the spatial IC's obtained for 3 different planes, z=14, 12, 8. All the crosshairs are placed on regions precisely located within visual cortex. The left image is is at voxel [24 9 14], x=12, y=-80, z=-15. The middle image is at [25 5 12], x=16, y=-96, z=5. The right image is at voxel [11 9 8], x=-40, y=-80, z=-15.

It can be assumed that the blobs in the lower region, the occipital lobe, correspond to the visual activity of interest, while the other independent components are related to other physiological processes, such as respiration, resting-state signals, etc., or to imaging artifacts, such as motion, ghosting, slice dropout, noise, etc. For example it can be conjectured that at z=14 the middle blobs may represent automatic movement control in the basal ganglia. The two blobs in the top most region may represent some type of executive functions occuring in the frontal lobe during the experiment.

The graphs on the following page show the plots of the actual IC's visualized here for each plane obtained by the FastICA algorithm.

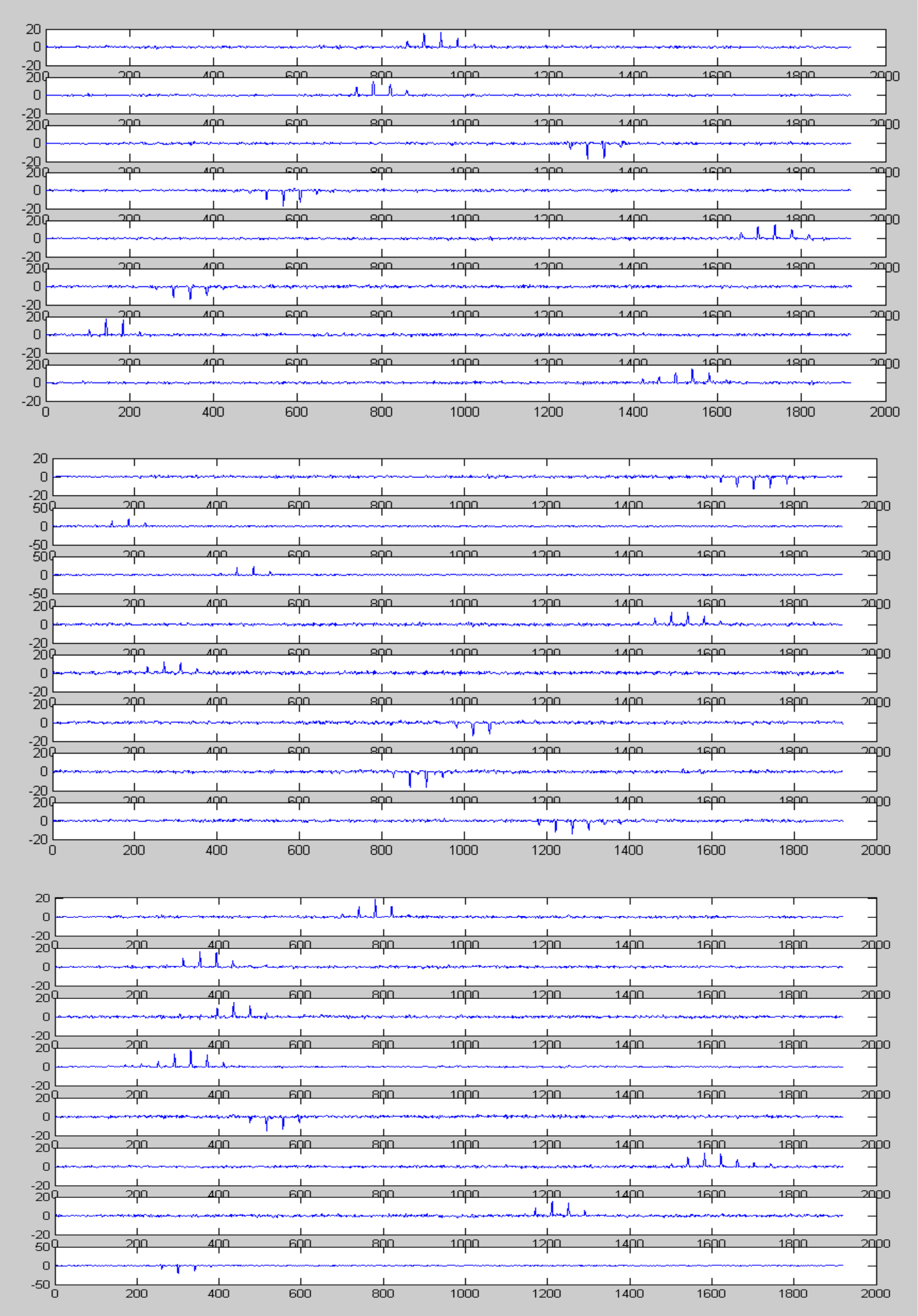


Fig 6. *8 spatial components extracted by FastICA algorithm for the 3 different tranches at z=14, 12, 8. Matrix of mixed signal consisted of one z-plane for all time signals. Thus rows consisted of vectorization of each z-plane. The means are not added in these plots.*

Table 3 *Significant regions of activity derived from ICA calculation for plane z=12 (5mm)*

Visual Cortex - Occipital lobe		Frontal, temporal lobe	
x, y, z (mm)	Voxel	x, y, z (mm)	Voxel
16, -96, 5	25 5 12	28, -24, 5	28 23 12
40, -88, 5	31 7 12	0, -12, 5	21 26 12
-52, -68, 5	8 12 12	4, 12, 5	22 32 12
		4, 40, 5	22 39 12
		8, 56, 5	23 43 12

The above table shows the energy values for all the IC's from the z-plane = 12. This particular plane was also chosen for the temporal decomposition analysis. Notice that ICA detected activity in the visual cortex as well as other regions in the brain during the visual funtion tests.

4.3 Spatial Components – Event-Related Design

The following images are the independent components computed from the event-related design experimental run taken at planes z=14, 12, 8. All the crosshairs are placed on regions located within visual cortex. The left image is is at voxel [23 6 14], x=8, y=-92, z=-15. The middle image image is at [23 6 12], x=8, y=-92, z=5. The right image is at voxel [24 8 8], x=-12, y=-84, z=-15. As compared to the IC's from the block designs, it appears that event-related type experiments do not generate as many IC's in the expected visual cortex region. It is yet unknown if the staccato type unfolding of stimuli presentation is the reason for the reduction of the efficacy of ICA.

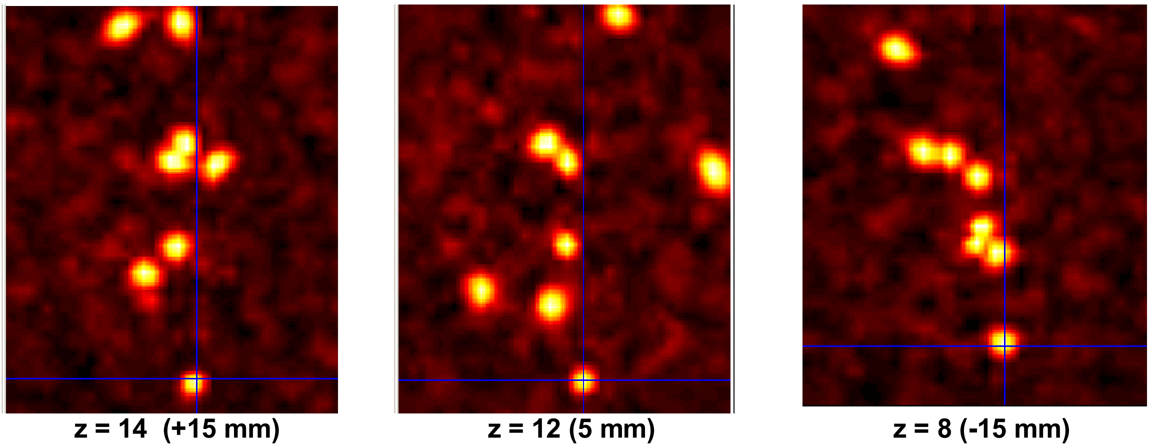


Fig 7. *8 spatial components from the event-related designed run extracted by FastICA algorithm on three different z planes visualized by SPM.*

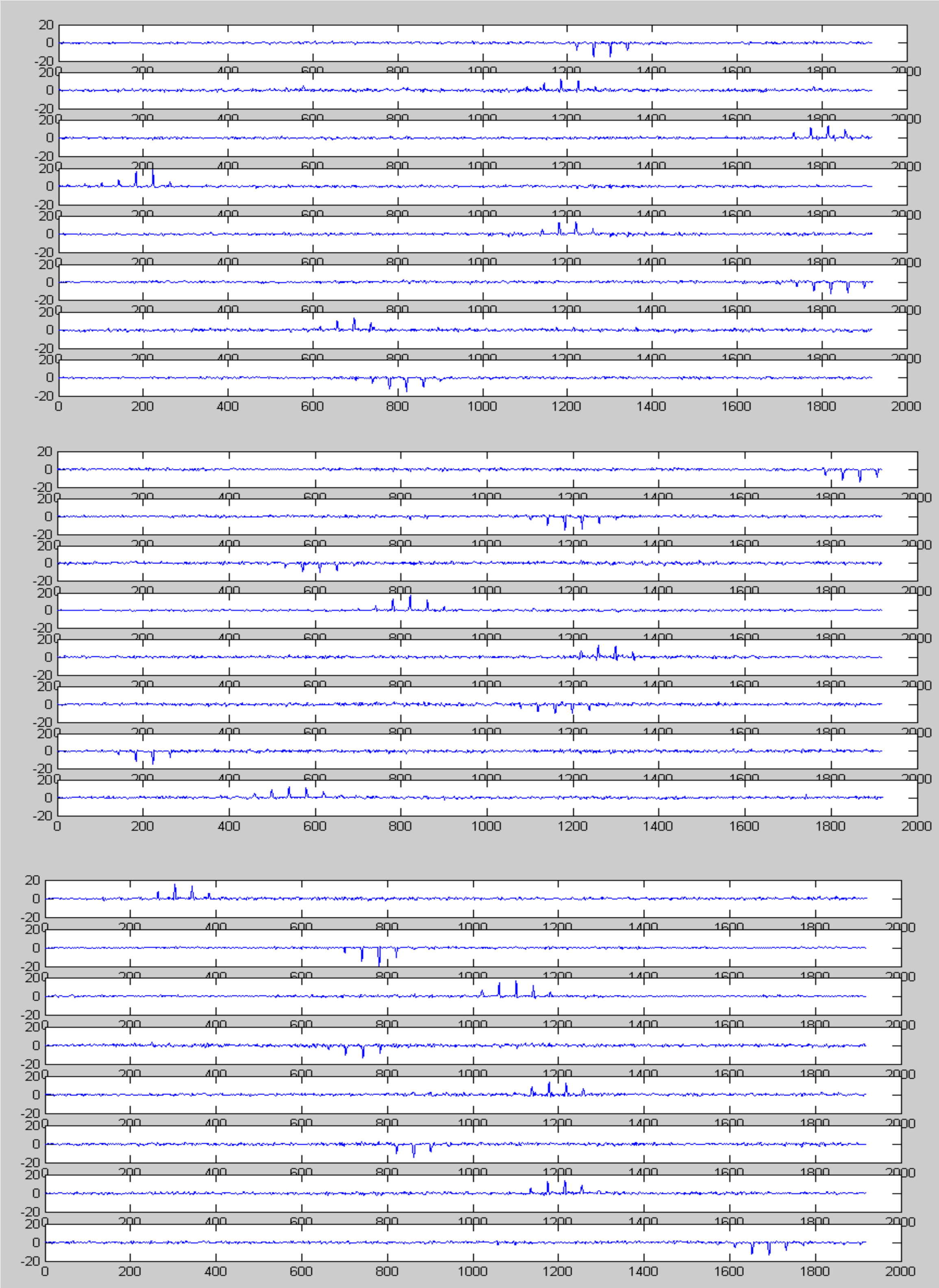


Fig 8. *8 spatial components extracted from the event-related design run by FastICA algorithm for the 3 different tranches at z=14, 12, 8. The means are not added in these plots.*

4.3 Temporal Components

As stated excessive noise and over-fitting was the concern in the analysis of the temporal components due to the sparseness of temporal points. Temporal calculations were attempted on the event-related design runs, but results contained excessive noise. Thus the results from the block design runs are included here. The transpose of the spatial matrix at the plane z=12 was taken and spatial vectors reduced to 15 voxel points from a total of 1920, centered around the voxel at [25 5 12]. This voxel region represents the very lower part of the right visual cortex.

The 3 graphs on the following page show 3 temporal independent components generated from calculations and the fourth graph is an average of all the time-courses. The blocks where color stimuli were presented is shaded to compare the time courses of the experiment with that predicted by ICA.

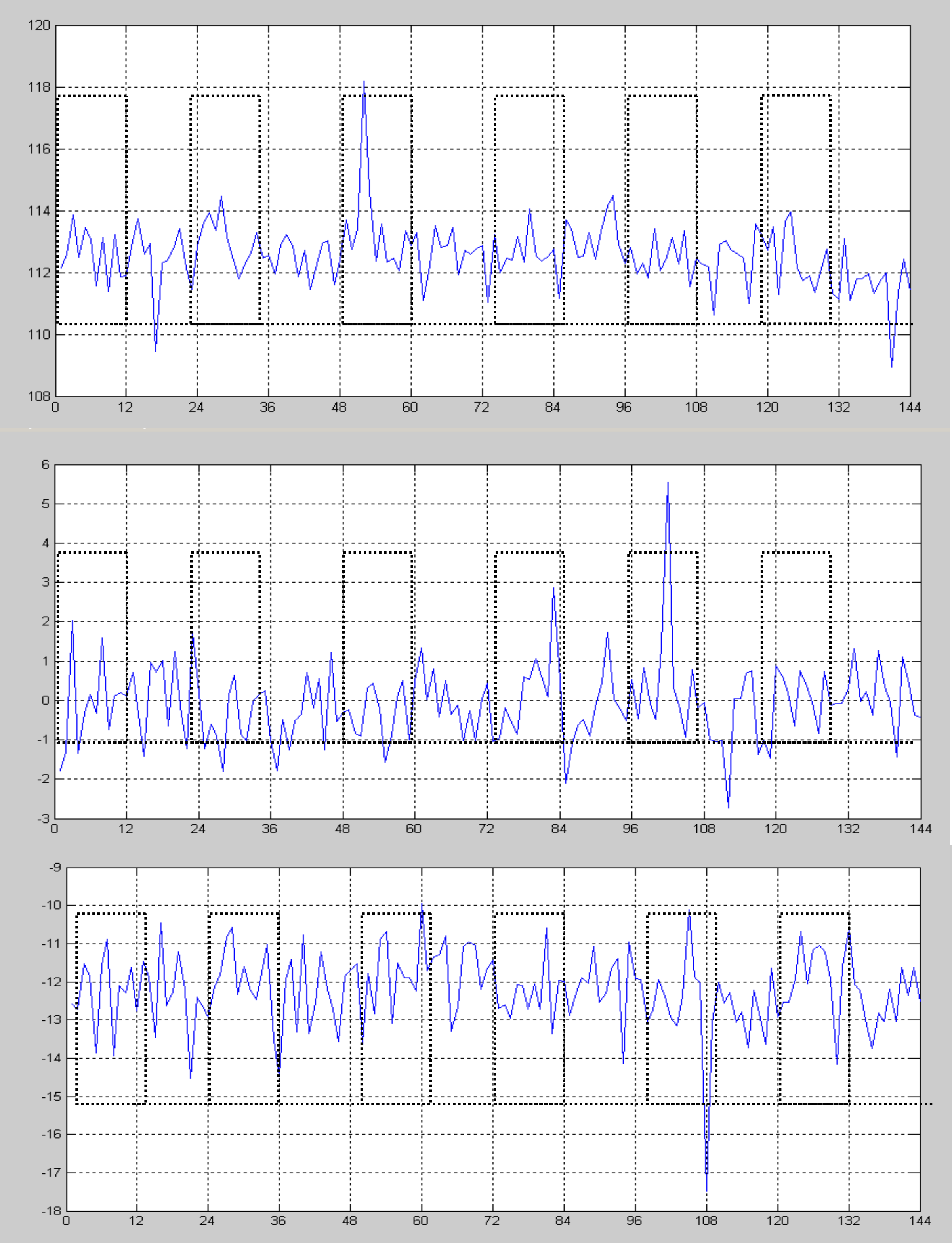


Fig 9. *Three different temporal components extracted from the lower visual cortex. Note the usually greater range of energy activity during periods of color stimuli as opposed to rest periods.*

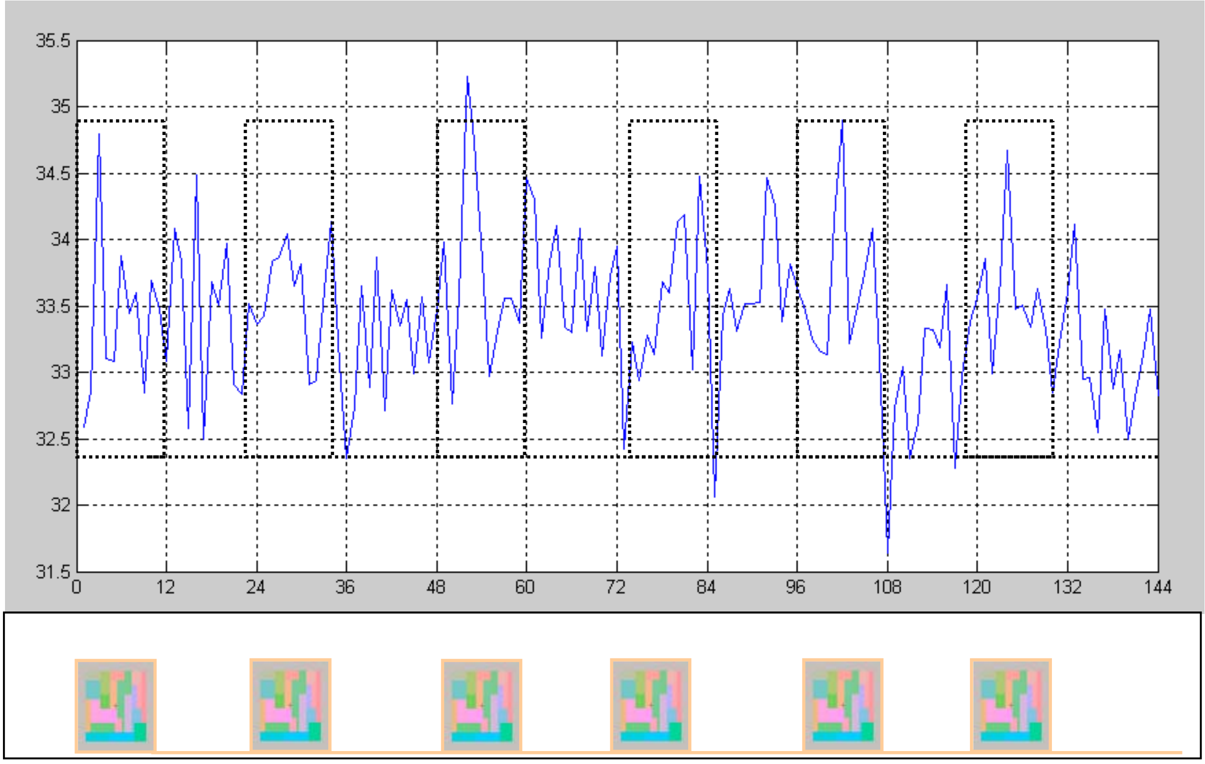


***Fig 10**. Average of the 3 previous temporal components compared to experimental task of alternating blocks of color and grey images. The temporal relationship is evinced: during periods of color stimuli the temporal components rise and then during rest periods of grey images activity falls.*

As expected there is heavy noise contained in the temporal components. Thus with the experimental and analytical procedures carried out in this study, correlation with the real task time courses would not be very strong. Nonetheless rising peaks can clear be seen within the time blocks of stimuli. In later time blocks one can begin to see anticipation of images by neuronal activity before the block change. One possibility to better measure correlation in future studies would be to correlate the amplitude spectrum of the temporal IC's.

It is estimated that the amplitude of the impulse response depends on the fraction of time spent in stimulation [10]. A design of further tests to support this idea could be to vary the block times. Again an event driven experiment would not be considered an ideal model for the testing of ICA due to the excessive noise generated and failure to distinguish between the extremely short transitions of stimulus to rest. Thus the minumum block lengths cannot be made too narrow.

Along these lines another research has suggested using the amplitude spectrum of the original fMRI data to calculate the IC's [5]. The motivation behind this idea is that one limitation of using ICA on fMRI is that the time course of any given component is required to have the same delay at every voxel. However different brain regions exhibit significant variations in hemodynamic delays. By using a model that allows for spatially varying delays (SVD), their research has showed that voxels with similar time courses, but different delays, are grouped into the same components.

The results from the temporal-spatial component analysis show that spatial ICA gives a more accurate measure of brain activity than do the temporal components. With some additional improvements in implementation and analytical procedure as stated, temporal ICA can offer valuable confirmatory analysis to hypothesis-driven models such as SPM. However when compared with such models, spatial IC's may provide additional insight. For instance it is yet unknown if the added IC's generated from regions outside the visual cortex is related in some way to visual functions which are not just artifacts or other unrelated physiological activity. It is not clear if GLM models are able to detect the additional IC's calculated from ICA. Further research is needed to see if these IC's bear an important relationship among specific cerebral functions where hypothesis-driven models such as SPM fail to discover.

Chapter 5

Conclusion

This study has shown that ICA is able to reveal hidden schemes within a system but certainly enough data points must exist for effective decomposition with minimal noise. The results from spatial analysis show more clearly the ability of ICA to finding underlying factors or sources to multivariate data.

Thus one area for improvement would be better temporal resolution. While enough spatial granularity existed for spatial decomposition to work well, insufficient time length prevented good temporal resolution. Practically speaking this may not always be possible since the timespan may be limited in actually keeping a person inside an MRI machine. It is not always a comfortable exercise, for example, to remain inside the catacomb atmosphere of the MRI machine. Thus a hybrid design of shortened and/or varied block times in order to increase the number of block variations in a run may help to increase the temporal resolution.

In future studies a comparison to the PICA algorithm (implemented in FSL-Melodic) may offer further insights into the robustness of the ICA methodology to fMRI data. One of the issues that arose in this study was the question of estimating the number of IC's to look for, which was not always clear. Whether some IC's were due to artifact or related to the vision experiment was unclear. The PICA algorithm implemented in FSL handles this problem by estimating the number of IC's and reducing dimensionality by discarding artifacts of non-interest.

In terms of the analysis of the IC's, while decomposition works well, interpretation of the actual independent components in regions outside of interest can be more subjective to the experimenter. For instance if activation is present in the frontal lobe area or parietal lobe, are these connected to the visual task at hand or are these artifacts?

Application of the ICA methodology to fMRI analysis in particular poses some interesting issues. ICA does not completely solve the inverse problem in that it produces an independent component analysis decomposition of the input data rather than an inverse source model decomposition.

From the perspective of the paradigm of ICA, brain sources are not necessarily viewed as localized fMRI activity, but rather concurrent temporal-spatial activity that is spatially fixed and independent of activity arising from other spatially fixed sources. The networks producing such concurrent activity are not due to local spatial distributions but rather by the covarying signatures they produce at the sensors. Thus the sources of ICA components may derive from distributed brain networks rather than physically localized active brain regions. Indeed these networks may be functionally linked, but one cannot determine that from the IC analysis.

The difference between these two views require a paradigm shift in the way one thinks about the brain. On the one hand the brain can be regarded as the collection of physically discrete neural networks which pass information to each other. This is the classic neurophysiologic viewpoint. Or the brain is a dynamically changing interaction of distributed, transient neural networks. This approaches the brain in terms of a dynamic system. Again it can be seen that the most fundamental questions as to the very nature of cognitve processes are breached even while trying to interpret the analysis of independent components of the brain.

Chapter 6

References

[1] McKeown, S. Makeig, G. Brown, T. Jung, S.Jindermann, A. Bell and T. Sejnowski, Analysis of fMRI data by blind separation into independent spatial components, Human Brain Mapping, 6:160-180 (1998).

[2] McKeown, M., Hansen, L., Sejnowski T., Independent Component Analysis of function MRI: what is signal and what is noise?, Current Opinion in Neurobiology, 13:620-629 (2003).

[3] Jutten, C. and Herault, J. Blind separation of sources, part 1: An adaptive algorithm based on neuromimetic architecture. Signal Processing, 24:1-10 (1991).

[4] Lalaroiu S. Kiviluoto K., Oja E., Times Series Prediction with Independent Component Analysis, Helsinki University of Technology.

[5] Calhoun, V., Adali, T., Pekar J., Pearlson, G., Latency (in)sensitive ICA Group independent component analysis of fMRI data in temporal domain, NeuroImage 20 (2003) 1661-1669.

[6] Ylipaavalniemi, J., Vigario, R., Analysis of Auditory fMRI Recordings via ICA: A Study on Consistency

[7] Svensen, M., Kruggel, F., Benali, H., ICA of fMRI Group Study Data, NeuroImage, 16, 551-563 (2002)

[8] Visscher, K, et. al., Mixed block/event-related designs separate transient sustained activity in fMRI, NeuroImage 18 (2003) 1694-1708

[9] Beckmann, C., Smith, S., Probabilistic Independent Component Analysis for Functional Magnetic Resonance Imaging, IEEE Transactions on Medical Imaging, Vol 23, No 2, February 2004

[10] R.M. Birn, P.A. Bandettini, Estimated BOLD impulse response depends on stimulus ON/OFF ratio, Human Brain Mapping, Brighton, (2001)

[11] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley & Sons, 2001.

[12] Nakai, T., et. al., Application of independent component analysis to magnetic resonance imaging for enhancing the contrast of gray and white matter, NeuroImage, 21: (2004) 251-260.

[13] Esposito, F., et. al., Real-time independent component analysis of fMRI time-series, NeuroImage, 20: (2003) 2209-2224.

[14] Chen, H., Yao, D., A composite ICA algorithm and the application in localization of brain activities, Neurocomputing, 56: (2004) 429-434.

Appendix A

Software Packages Used

The FastICA package can be found at:
http://www.cis.hut.fi/projects/ica/fastica

The SPM package can be found at:
http://www.fil.ion.ucl.ac.uk/spm/

The FSL package can be found at:
http://www.fmrib.ox.ac.uk/fsl/

Matlab Code

```
% Code used to analyse fMRI data
% calls functions to SPM and FastICA
%
% -------------------------
% Extract Spatial Composants
% -------------------------
% Using spm2 functions

% spm_vol
% spm_sample_vol
% spm_slice_vol

% Must be in ../ica/aworkblockX/
% Expect img and hdr files in ../donne/blockYYY/

minTranche = 1;
maxTranche = 20;
minN = 1;
maxN = 40;  % horizontal
minM = 1;
maxM = 48;  % vertical

if spmVer == '2'
for h = 1:maxObs
  fileRoot = [filePrefix, int2str(h)];
  fileSource = [fileRoot, '.img'];
  fileFull = [dirDonnee, fileSource];
  % Create a handle to the image volume.
  V = spm_vol(fileFull);
  disp(fileFull);
  for i = minTranche:maxTranche;
    % Define which voxels you want to get at.  e.g., plane 10...
    [x,y] = ndgrid(1:V.dim(1),1:V.dim(2));
    z = ones(size(x))*i;    % 40x48 of iSlice
    % sample the data, get voxel timecourses
    dataSample = spm_sample_vol(V,x,y,z,0);
    outputFile = [fileRoot, '_s', int2str(i)];
    dlmwrite(outputFile, dataSample, '\t');
    outMsg = [fileRoot, '- Tranche: ', int2str(i)];
    disp(outMsg);
  end
end

else  % Depricated spm99 code!
% ----------------------------
timeVoxel = [];
for h = 1:maxObs
  fileRoot = [filePrefix, int2str(h)];
  fileSource = [fileRoot, '.img'];
  fileFull = [dirDonnee, fileSource];
  % [DIM VOX SCALE TYPE OFFSET ORIGIN DESCRIP]=spm_hread(fileFull)

  % pour recuper une matrice de donnees
  info = spm_vol(fileFull);
  volMtx = spm_read_vols(info);
  % outputFile = [fileRoot, '_s', int2str(i)];
  % dlmwrite(outputFile, dataSample, '\t');

  % Vectorize for each t ...
  vect = [];
  %for iN = 1:maxN  %40
  %  vect = [vect volMtx(iN,:,Tranche)];
  %  outMsg = ['row: ', int2str(iN), ' of ', fileSource];
  %  disp(outMsg);
  %end

  for iM = 1:maxM  %48
    vect = [vect volMtx(:,iM,Tranche)'];
    outMsg = ['row: ', int2str(iM), ' of ', fileSource];
    disp(outMsg);
  end

  timeVoxel = [timeVoxel; vect];
end

end
return;
```

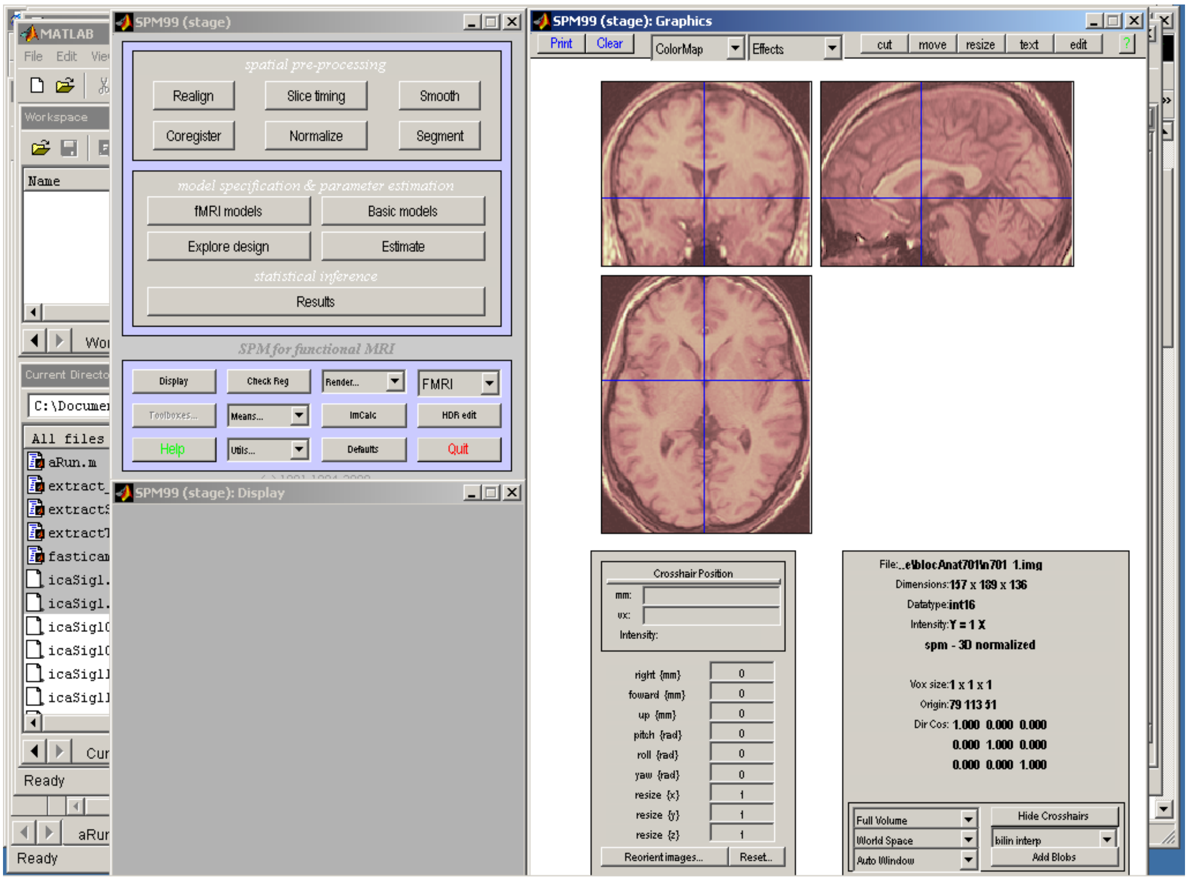
```
fid=fopen(nameFile,'r');
info=spm_vol(nameFile);
data=zeros(info.dim(1),info.dim(2),info.dim(3));
for i=1:info.dim(3)
data(:,:,i)=fread(fid,info.dim(1:2),spm_type(info.dim(:,4)));
end
fclose(fid);
    dlmwrite(outputFile, dataSample, '\t');

return;



% -----------------
% Vectorize Volumes
% -----------------

% Vectorize - pre-processing before fastica routine
% Soit vol block, soit plane
% snr301_1_s10 - format of filename from extractBlock.m

fileRoot = 'snr601_';
maxObsv = 144;
dirDonnee = '';

% Determine region u want to cut out
minTranche = 10;
maxTranche = 10;
minN = 1;
maxN = 40;  % horizontal
minM = 1;
maxM = 48;  % vertical

for iT = minTranche:maxTranche;
  slc = ['s', int2str(iT)];
  % gotta change this to array.....
  timeVoxel = [];

  for t = 1:maxObsv;
    fileSource = [fileRoot, int2str(t), '_', slc];
    fileFull = [dirDonnee, fileSource];
    snrMtrx = load(fileSource);
    disp(fileSource);
    % Vectorize for each t ...
    vect = [];
    % for iN = maxN:-1:1;
    for iN = 1:maxN
      vect = [vect snrMtrx(iN,:)];
      % dlmwrite(outputFile, dataSample, '\t');
      outMsg = ['row: ', int2str(iN), ' of ', fileSource];
      disp(outMsg);
    end
    timeVoxel = [timeVoxel; vect];
    clear fileSource;
  end
end


% --------------------
% Run fastica manually
% --------------------

mixedSig = timeVoxel;

[icaSig, A, W] = fastica(mixedSig,'numOfIC', maxIC, 'approach',
'defl', 'g', 'pow3');

%[icaSig] = fastica(mixedSig, 'lastEig', 30, 'numOfIC', 10);
%               Reduce dimension to 10, and estimate only 3
%               independent components.
% figure;
% icaplot('classic', icaSig)

% Reverse row
%[rMax cMax] = size(icaSig);
%icaTemp = icaSig;
%for k = 1:cMax
%  icaSig(:,k) = icaTemp(:,cMax - k + 1);
%end

return;



% -------------
% ICA to Matrix
% -------------

% Convert IC's to maxtrix form for spm
% extractBlock & vectorize
% fasticam
% icaToMtx

%spmVer = '99';
%maxIC = 15;
%SampleImgFile = 'snr801_5.img';
%Tranche = 10;
%maxN = 40;
%maxM = 48;

minTranche = 1;
maxTranche = 1;
maxZ = maxTranche + 1 - minTranche;
[maxIC, maxLen] = size(icaSig);

% Reconstruct back to 40x48 matrix
```

```
icaMtxOut = ones(maxN,maxM,maxZ,maxIC);
icaMel = ones(maxN,maxM,maxZ);
icaNorm = ones(maxIC,maxLen);
% For old versions, no minmax: 6.1.0.450 (R12.1)
% pr = minmax(icaSig);
% start R12 minmax routine
pmin = min(icaSig');
pmax = max(icaSig');
pr = [pmin', pmax'];
% end R12 minmax routine
for iC = 1:maxIC
  if abs(pr(iC,1)) > abs(pr(iC,2))
    %icaNorm(iC,:) = 1;
    icaNorm(iC,:) = -1*icaSig(iC,:);
    shift = ceil(1 - min(icaNorm(iC,:)));
    icaNorm(iC,:) = icaNorm(iC,:) + shift;
    fact = 1000/max(icaNorm(iC,:));
    icaNorm(iC,:) = icaNorm(iC,:)*fact;
  else
    %icaNorm(iC,:) = 1;
    icaNorm(iC,:) = icaSig(iC,:);
    shift = ceil(1 - min(icaNorm(iC,:)));
    icaNorm(iC,:) = icaNorm(iC,:) + shift;
    fact = 1000/max(icaNorm(iC,:));
    icaNorm(iC,:) = icaNorm(iC,:)*fact;
  end
  for iZ = minTranche:maxTranche

    %iM = 1;
    %for iN = 1:maxN
    %  stepLen = iM+maxM-1;
    %  icaMtxOut(iN,:,iZ,iC) = icaNorm(iC,iM:stepLen);
    %  outMsg = ['IC: ', int2str(iC), ' Row N: ', int2str(iN), '
Col M: ', int2str(iM), ' to ', int2str(stepLen) ];
    %  disp(outMsg);
    %  iM = iM + maxM;
    %end

    iN = 1;
    for iM = 1:maxM
      stepLen = iN+maxN-1;
      icaMtxOut(:,iM,iZ,iC) = icaNorm(iC,iN:stepLen)';
      outMsg = ['IC: ', int2str(iC), ' Row N: ', int2str(iN), '
Col M: ', int2str(iM), ' to ', int2str(stepLen) ];
      disp(outMsg);
      iN = iN + maxN;
    end

  end
end

figure;
icaplot('classic', icaNorm);
```

```
% ----------------------
% Write out to img format
% ----------------------
Vin = spm_vol(SampleImgFile);  %'snr301_10.img'
Vout = Vin;
Vout.dim(3) = 1;

% Write out using new spm2 functions
% ---------------------------------
% i = findstr(Vin.fname,'/');
% Vout.fname = [Vin.fname(1:i(end)),'d',Vin.fname(i(end)+1:end)];
% Data = spm_read_vols(Vin);
for iC = 1:maxIC
  Vout.fname = ['icaSig_T', int2str(Tranche), '_S', int2str(iC),
'.img'];
  spm_write_vol(Vout,icaMtxOut(:,:,1,iC));
end

% Combine IC's to single matrix and img
for iC = 1:maxIC
  icaMel = max(icaMel,icaMtxOut(:,:,1,iC));
end
Vout = Vin;
Vout.dim(3) = 1;
Vout.fname = ['icaSigMel_T', int2str(Tranche), '_IC',
int2str(maxIC), '.img'];
spm_write_vol(Vout,icaMel);

return;

% Write out using depricated spm99 fns
% -----------------------------------
OutputMelFile = 'icaSigMel99.img';
[DIM VOX SCALE TYPE OFFSET ORIGIN DESCRIP] =
spm_hread(SampleImgFile);
DIM(3) = 1;
fid = fopen(OutputMelFile,'wb');
fwrite(fid,icaMel,spm_type(TYPE));
fclose(fid);

return;



% --------------------------
% Extract Temporal Composants
% --------------------------

% spm_vol
% spm_sample_vol
% spm_slice_vol
% Must be in ../ica/aworkblockX/
% Expect img and hdr files in ../donne/blockYYY/

minN = 1;
maxN = 40;  % horizontal
minM = 1;
maxM = 48;  % vertical

timeVoxel = [];
for h = 1:maxObs
  fileRoot = [filePrefix, int2str(h)];
  fileSource = [fileRoot, '.img'];
  fileFull = [dirDonnee, fileSource];
  % [DIM VOX SCALE TYPE OFFSET ORIGIN DESCRIP]=spm_hread(fileFull)

  % pour recuper une matrice de donnees
  info = spm_vol(fileFull);
  volMtx = spm_read_vols(info);
  % outputFile = [fileRoot, '_s', int2str(i)];
  % dlmwrite(outputFile, dataSample, '\t');

  % Vectorize for each t ...
  vect = [];
  %for iN = 1:maxN  %40
  %  vect = [vect volMtx(iN,:,Tranche)];
  %  outMsg = ['row: ', int2str(iN), ' of ', fileSource];
  %  disp(outMsg);
  %end

  for iM = TM_beg:TM_end
    vect = [vect volMtx(TN_beg:TN_end,iM,Tranche)'];
    outMsg = ['Temporal iM: ', int2str(iM), ' TN_beg: ',
int2str(TN_beg), '  TN_end: ', int2str(TN_end), '  file: ',
fileSource];
    disp(outMsg);
  end

  timeVoxel = [timeVoxel; vect];
end

timeVoxel = timeVoxel';

return;


% -------------------
% Output ICA Temporal
% -------------------

[maxIC, maxTmp] = size(icaSig_Tmp);
icaPd_Tmp = sum(icaSig_Tmp)/maxIC;
% icaPd_Tmp = icaSig_Tmp(ICAsmpl,:);
```

```
figure;
plot(icaPd_Tmp);
set (gca,'XLim',[1 144]);
set (gca,'XTick',[1:11:144]);
grid on


% --------------
% Master program
% --------------


% ----------------
% Global variables
% 0 = spatial components
% 1 = temporal components

routine = 1;
spmVer = '99';
maxObs = 144;
dirDonnee = '../donnee/bloc801/';
filePrefix = 'snr801_';
Tranche = 8;

maxIC = 6;  % first used by fasticam

SampleImgFile = 'snr801_5.img';
maxN = 40;
maxM = 48;
maxObsv = 144;

TN_beg = 5;
TN_end = 20;
TM_beg = 9;
TM_end = 9;

% end global variables
% --------------------

if routine == 0
  % Main routines for Spatial IC
  % ----------------------------
  disp('Running SPATIAL analysis');
  extractSpatial;
  %vectorizeVol;
  fasticam;
  icaSig_Spt = icaSig;
  A_Spt = A;
  W_Spt = W;
  icaToMtx;
else
  % Main routines for Temporal IC
  % -----------------------------
  disp('Running TEMPORAL analysis');
  extractTemporal;
  fasticam;
  icaSig_Tmp = icaSig;
  A_Tmp = A;
  W_Tmp = W;
  icaTemporal;
end

return;


spmVer = '2';
sampleFileImg = 'snr301_10.img';
outputFile = 'ica_snr301_10_s10.img';
SampleImgMtx = 'snr301_10_s10'
load snr301_10_s10;

if (spmVer == '2')
Vin = spm_vol(sampleFileImg);
Vout = Vin;
% i = findstr(Vin.fname,'/');
% Vout.fname = [Vin.fname(1:i(end)),'d',Vin.fname(i(end)+1:end)];
Vout.dim(3) = 1;
% Data = spm_read_vols(Vin);

%for iC = 1:8
  %Vout.fname = ['icaSig', int2str(iC), '.img'];
  Vout.fname = outputFile;
  spm_write_vol(Vout,snr301_10_s10);
% end

else

[DIM VOX SCALE TYPE OFFSET ORIGIN DESCRIP] =
spm_hread(SampleImgMtx);
fid = fopen(outputFile,'wb');
fwrite(fid,snr301_10_s10,spm_type(TYPE));
fclose(fid);

end
```

Appendix B



Ready

Ready

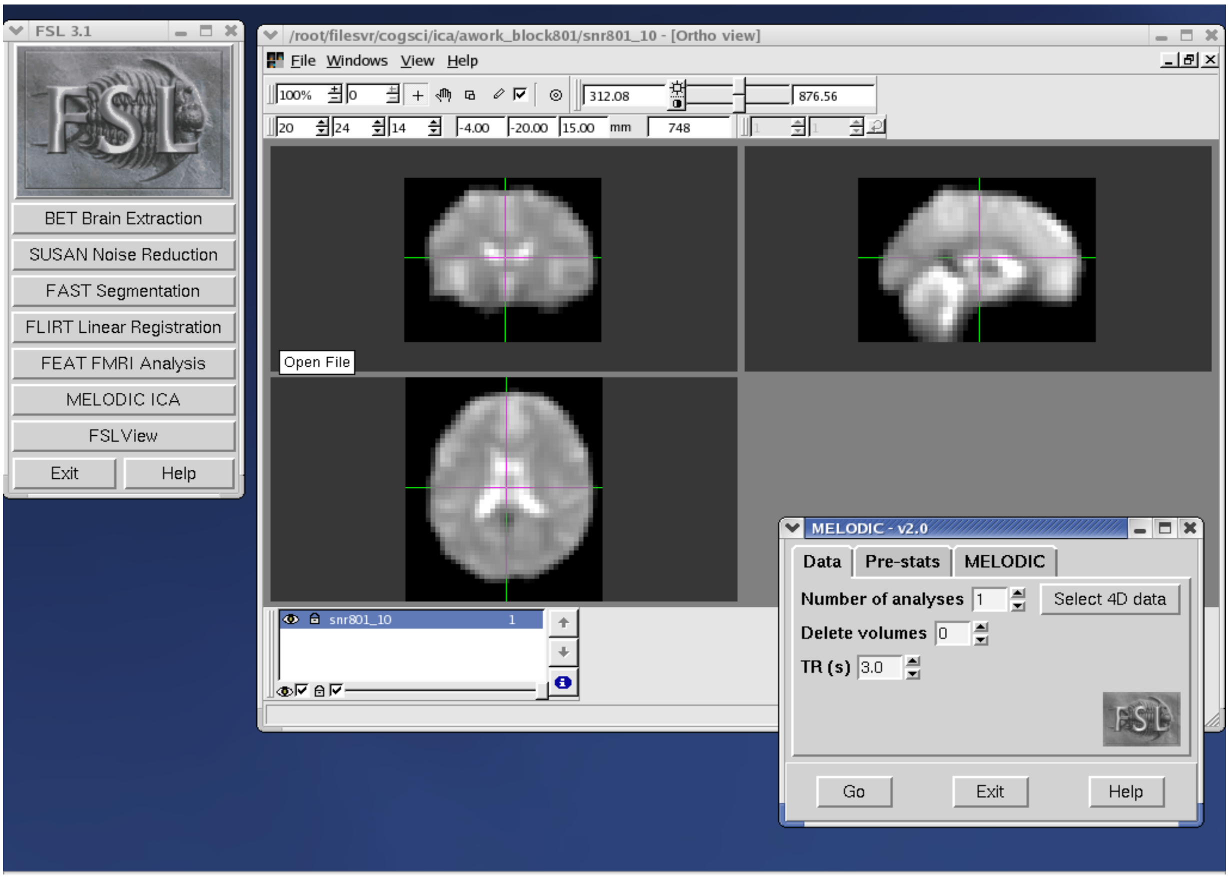
right {mm} 0
foward {mm} 0
up {mm} 0
pitch {rad} 0
roll {rad} 0
yaw {rad} 0
resize {x} 1
resize {y} 1
resize {z} 1

File:..e\blocAnat701\n701 1.img
Dimensions:157 x 189 x 136
Datatype:int16
Intensity:Y = 1 X
spm - 3D normalized

Vox size:1 x 1 x 1
Origin:79 113 51
Dir Cos: 1.000 0.000 0.000
0.000 1.000 0.000
0.000 0.000 1.000

Full Volume
World Space
Auto Window
bilin interp

Fig 9. Screenprint of SPM, ver 99, on Windows platform.



Number of analyses 1

Delete volumes 0

TR (s) 3.0

Fig 10. Screenprint of FSL from Oxford Centre for Functional Magnetic Resonance Imaging of the Brain of Oxford University on Linux platform. Provides similar image functionality to SPM, but adds ICA estimation routines using PICA algorithm based on Bayesian analysis.