Faculty of Mathematics and Natural Sciences

Department of
Mathematics and
Computing Science

Wavelet-based Statistical Analysis of MRI Images

G. Loots

Advisor:
dr. J.B.T.M. Roerdink

December, 1999

RuG

Wavelet-based Statistical Analysis of MRI Images

G. Loots

December 21, 1999

Contents

List of Figures

Chapter 1

Introduction

There are quite a number of programs available for detecting and visualizing brain activation. From the medical side there exists a large interest in these programs. In a cooperation between the academic hospital of Groningen, the Netherlands (AZG), and the university of Groningen (R*u*G), a research program called GNIP has been founded (Groningen NeuroImaging Project). One of the activities of the GNIP project is to investigate the use of these existing programs, and to make a comparison of their advantages and disadvantages. Another research interest is to investigate other approaches of studying brain activation, and implement other methods to obtain possible new insights in brain visualization. The approach which is presented in this thesis, is based on a combination of wavelets and a statistical framework which is applied in the wavelet domain. A wavelet based analysis is expected to have a better signal-to-noise ratio (SNR) than a conventional spatial analysis, because there is no need to smooth (i.e. filter) the presented data as is usually done in spatial analysis. Wavelets provide a multiresolution method to minimize noise, which means that the noise is expected to be located at different decomposition levels and therefore can be dealt with accordingly at each level. The wavelet method derives its power from the fact that a smooth and spatially localized signal can be represented by a small set of wavelet coefficients, whereas the power of white noise is uniformly spread throughout the wavelet space. The data which have been used in this thesis are fMRI data which have been provided by the AZG.

Chapter 2

Concepts of brain activation studies

An important purpose of brain activation studies is to examine at which physical location in a brain (which can be human or non-human) an increase in activity can be measured when the subject performs certain tasks. For analysis, a series of scans is taken of the brain of the subject. The data-acquisition can be done at hospitals or research facilities using quite a number of modalities, which will be described here shortly. The scans are usually made in two conditions, and the objective is to find significant differences between the scans corresponding to different conditions.

2.1 Modalities

Images of a brain can be obtained through different techniques. The three most important ones are listed in the next subsections.

2.1.1 Computed Tomography (CT)

This technique uses X-rays to acquire information. A Röntgen tube circulates around the subject, and records several thousands of projections. The X-rays are picked up by an array of sensors. Using various computational techniques (see [13]), an image is reconstructed. Several slices can be combined to produce a three-dimensional image. The images will show the density of the scanned region. Especially structures of bone are well-detected using this scanning technique. These bone structures show up white (high density) on the acquired images. Since this technique uses X-rays, it has to be much limited in both time and resolution, in case a living subject is examined. If a subject is exposed to too radiation, then the subject will be harmed. Figure 2.1 shows an example of a CT image.

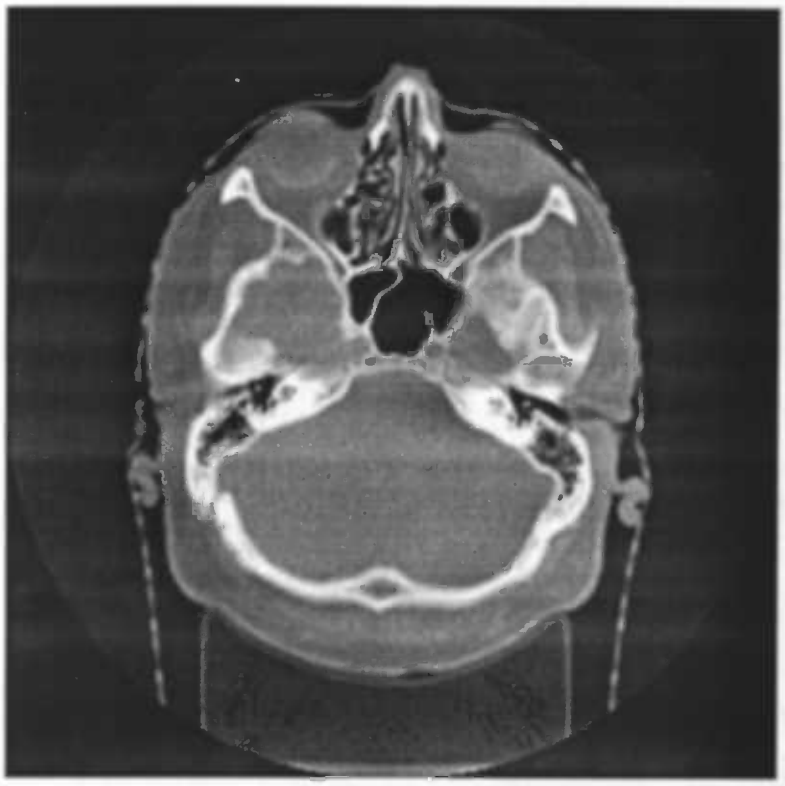

Figure 2.1: CT image

2.1.2 Positron Emission Tomography (PET)

This technique involves injecting a subject with a radio-labeled biologically active compound, called a tracer, which decays through the emission of a positron particle. This particle annihilates with an electron from the surrounding tissue, emitting 2 gamma rays which are detected in coincidence by a scintillation gamma camera. Once the data is collected, special algorithms and computer programs produce a 3D image of the subject's distribution of physiological processes. Interpreting these images can be pretty hard for an untrained person, because most of the PET images do not show anatomic information such as, for example, bone tissue. Figure 2.2 shows an example of a PET image.

2.1.3 Functional Magnetic Resonance Imaging (fMRI)

In the early of the 80s the PET modality had a dominating position in functional anatomy. fMRI has only recently made a breakthrough, and has developed into an alternative and powerful technique. PET measures blood flow on a spatial scale of about 6 mm and a temporal scale of approximately 30 seconds. Functional MRI measures blood-oxygenation level dependent (BOLD) signal changes caused by

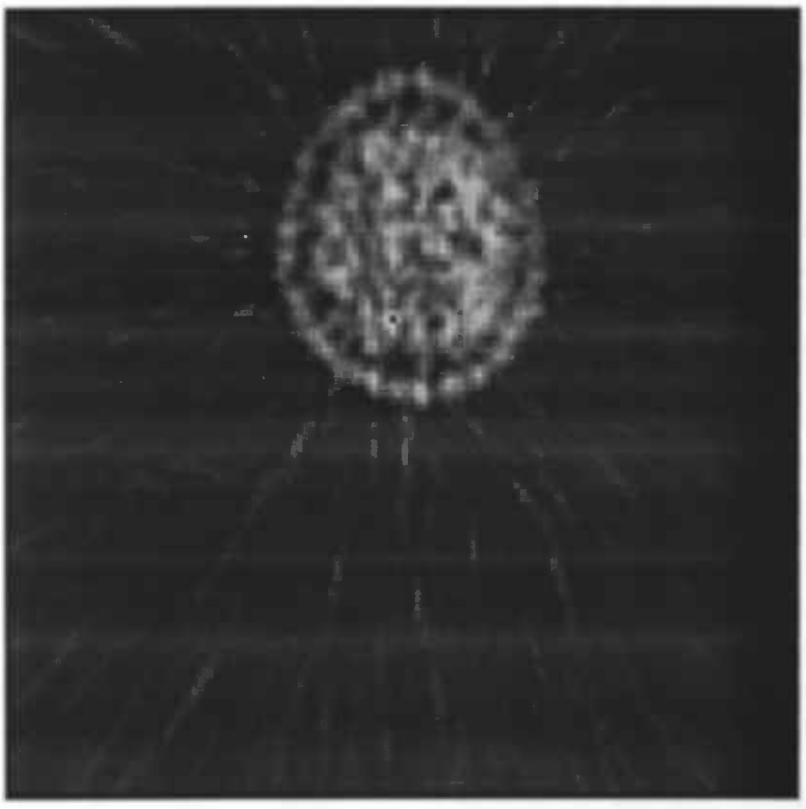

Figure 2.2: PET image

regional hemodynamic adjustments in response to changes in neuronal activity [14]. The current understanding is that an increase in brain activity leads to hyperoxemia (a decreased level of deoxyhemoglobin), which is due to an over-compensation of the local tissue perfusion in response to the increased energy demand in the activated neurons [8]. However, the exact mechanisms of these regulatory processes are not fully known. In particular, the interpretation of BOLD signal decreases is not yet established. Nonetheless, BOLD methods offer considerable advantages over other functional modalities, in that they can be performed on widely available clinical scanners, do not require exogenous contrast agents or exposure to ionizing radiation, provide excellent spatial resolution, and can be registered with anatomical images acquired on the same machine. Soft tissue shows up very well on MRI images. Bone tissue shows up black, which is opposite behavior when compared to CT imaging. Figure 2.3 shows an example of a MRI image.

Although the first reports of imaging in humans based on BOLD effects appeared in 1992 [10], important issues regarding sensitivity, reproducibility, and the nature of artifacts are still not settled. Neuronal activity changes induced by various experimental stimuli typically result in signal intensity changes of 1%–5% in

1.5-T scanners, which are close to the scan-to-scan variability. While SNR can often be improved by stimulus repetition with subsequent scan averaging, there is a practical number of scans that can be collected in a single human subject. Changes in physiological processes as a result of habituation, learning or fatigue, subject motion, and machine calibration drift, impose time constraints on the duration of an experiment. Consequently, scan acquisition time is an important factor. A major problem of BOLD methods is the presence of artifacts associated with head and/or vessel motion [5], as well as vascular inflow and drainage effects. In attempts to overcome some of these technical difficulties, new image acquisition schemes are rapidly evolving where, however, formal performance comparisons often are not presented [15].

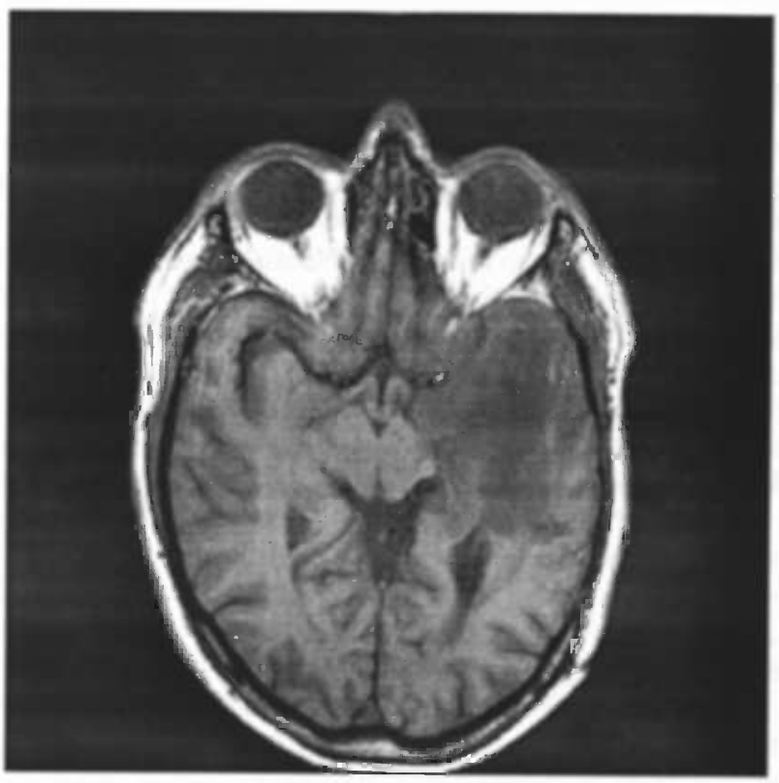

Figure 2.3: MRI image

2.2 Required stages for data analysis

2.2.1 Realignment

Due to the movement of a subject in a scanner, the data can be distorted. This movement-related variance has to be removed to make it possible to properly

analyze the data. For PET the movement, relative to the first scan, is estimated using a least squares method that is obtained after linearizing the problem using a first order Taylor series. The realignment then simply involves re-slicing the scans, using the motion estimates.
fMRI time-series constitute a different case. This is because movement in earlier scans can affect the signal in subsequent scans (due to differential spin-excitation histories). The realignment for fMRI time-series consists of two parts. First the images are realigned, just like the PET-data. Second there is a mathematical adjustment based on a moving average auto-regression model of spin-excitation history effects. For realignment, a package called `imreg` has been used. The theoretical aspects of how this package works are explained in section 5.1. of this thesis.

2.2.2 Spatial smoothing

Smoothing has a number of important objectives. In the first place, it generally increases signal relative to noise. This is because noise usually has higher spatial frequencies than the neuro-physiological effects of interest, which are produced by hemodynamic changes that are expressed over spatial scales of several millimeters. The second reason for smoothing is specific to inter-subject averaging. It ensures that hemodynamic changes from subject to subject are assessed on a spatial scale at which homologies in functional anatomy are typically expressed. In other words it's unlikely that inter-subject brain activity will match on a scale of much less than a millimeter. But it is much more likely that it will match on a scale of say 8 mm, evidenced by the success of multi-subject PET activation studies. The final reason is that the data conforms more closely to a Gaussian field model after smoothing. This is important, because the statistical inference makes use of the theory of Gaussian fields. With smoothing there is a trade-off between the degree of smoothing and the spatial resolution to be considered. This trade-off is especially important for fMRI with its high anatomical resolution. The general approach is to use the least possible smoothing, but to have a smoothness being large enough to ensure the correctness of the statistical inference. Note that the use of spatial smoothing is the subject of intensive scientific debate (see also [6], [4] and [9]).
By their multi-scale nature, wavelet-based methods constitute a promising new approach to this problem, see below.

2.2.3 Statistics

In this final stage the obtained volume scans are used as an input for various statistical tests. The objective is to show which part of the brain is significantly different, when compared to another series of scans.

2.3 Common experimental approach

A quite common approach to examining brain activation of a human subject is to acquire scans of the subject under different conditions. For example, the subject can be presented an audible or visible signal during the acquisition of the first series of scans, and be left in a rest state during the second series of acquisitions. Then finding the activation patterns in the brain is accomplished by finding significant differences between the images that were made in the first and second series of scans.
However, the differences (if any) are usually extremely subtle and small. Therefore a statistical approach for analyzing the data is a necessity, because a difference image of 2 images (which were not acquired during the same setup) hardly reveals a visibly noticeable pattern.
Replicating the same experiment tends to improve the detectability of a significant signal. This is because it offers the opportunity to eliminate more noise: if one assumes that the noise in the input images is white, then the average of a large number of such images will contain less noise. However, just acquiring more and more images is definitely not the only goal to go for in these studies, because from practice it can be learned that there also exists noise that seems to be correlated to the absence or presence of the stimuli. That particular noise will stay present in average images. Therefore, a statistical model was developed [16] in the wavelet domain. A major advantage of a wavelet approach is the fact that such a strategy is based on a multi-resolution analysis, and therefore it can deal with noise at multiple scales ([16], see also chapter 3).
A widely used concept in such experiments is the use of a block-design. This is a experiment setup in which where a certain amount of scans in the same class is called a *block*, and the experiment is done with different blocks following each other (for example, if an *on*-block is represented with a '1' and if an *off*-block is represented with a '0', a possible block setup can be '101010'. When an experiment has been replicated N times, the first step in the data-analysis process is calculating N block difference images (this is accomplished by subtracting the means of the images acquired within 'on'-blocks from the means of the images within 'off'-blocks), by

$$g_i(n) = g_i^{(1)}(n) - g_i^{(0)}(n), \quad i = 1, \ldots, N, \tag{2.1}$$

where $n \in \mathrm{Z}^q$ is the equidistant sampling grid on Z^q, and where $g_i^{(1)}(n)$ and $g_i^{(0)}(n)$ are images acquired during the presence of a stimulus c.q. absence of a stimulus, respectively. In most fMRI applications, q usually has the value 2 or 3.
The difference images are assumed to be characterized by the population model

$$g_i(n) = f(n) + e_i(n), \tag{2.2}$$

where $f(n)$ is the unknown deterministic signal (common to all replications), which one would like to recover, and e_i is a homogeneous random field of iden-

tically and independently distributed Gaussian noise, $iid \sim \mathcal{N}(0, \sigma^2)$. Averaging over N uncorrelated observations improves the SNR and yields a signal estimate:

$$\hat{f}(n) = \frac{1}{N} \sum_{i=1}^{N} g_i(n) \tag{2.3}$$

which is asymptotically unbiased and consistent ($N \to \infty$); i.e.

$$\mathrm{E}[\hat{f}(n)] = f(n), \qquad \mathrm{Var}[\hat{f}(n)] = \sigma^2/N. \tag{2.4}$$

The estimated sample variance at each pixel location n

$$\hat{\Sigma}^2(n) = \frac{1}{N-1} \sum_{i=1}^{N} [g_i(n) - \hat{f}(n)]^2 \tag{2.5}$$

has $(N-1)$ degrees of freedom (DOF's).
The use of the previous formulas requires the assumption of uncorrelated replications of the random field, which was sufficiently well satisfied for the mean block images $g_i^{(1)}$ and $g_i^{(0)}$, because of the relatively long duration of the block cycle. This assumption would certainly be violated if the volumetric images at each time point in the acquisition series were considered as replications, due to their high temporal correlation. If a time-series variance were to be used for statistical inference in the spatial domain, either specific time-series analysis methods that properly adjust for temporal correlation needed to be applied, or the effective DOF's for the residual variance in a general linear model incorporating the hemodynamic response function needed to be estimated.
Since e_i is homogeneous by assumption, an improved variance estimate can be obtained by pooling $\hat{\Sigma}^2(n)$ over all intracranial (IC) pixels (those pixels which are located within the brain volume), $n_{\mathrm{pix}} = \#\mathrm{IC}$, yielding an approximation of σ^2 with very large DOF:

$$\sigma^2 = \frac{1}{n_{\mathrm{pix}}} \sum_{n \in \mathrm{IC}} \hat{\Sigma}^2(n). \tag{2.6}$$

The recovery procedure of $f(n)$ is then cast within the framework of hypothesis testing. The null hypothesis H_0 postulates $f(n) = 0$; i.e. there is no systematic difference between the images acquired under the two different experimental conditions. If the hypothesis is refuted by the data, then the inference is that the signal is nonzero at certain spatial locations. In that case, it is of particular interest to the user of the system to obtain both a good estimate of the spatial locations and shape of the signal at these locations.
A method to apply this statistical framework is to assume that the images in question can be approximated by a continuous random field, where the pixel values are considered to be the realizations of a random field sampled on an equally

spaced lattice n. The relevant test statistics are then evaluated at each pixel and searched for local extrema that might indicate the presence of an activation signal. Formal statistical methods have been developed to guard against false positive detections, which provided explicit expressions for the probability of excursion sets of Gaussian, $t-$, χ^2-, and $F-$tests [12]. However, while these methods are quite elegant, potential drawbacks of the random field approach are that

a. a smoothness parameter (usually FWHM, the full width half maximum) of the point-spread function of the imaging method is required to be either known or imposed by filtering

b. the images be sufficiently finely sampled (FWHM/pixel size) > 2.

Unfortunately, the second condition does not hold for fMR images, and with regard to the first condition, the 'proper' amount of further smoothing to be applied is often inextricably related to the research question itself. Since random field methods cannot be used in the current context (at least not without prior image smoothing), the results of the wavelet based analysis developed below will be compared to results obtained by a spatial domain analysis, where the false positive detection rate (i.e. the significance level) is controlled by the Bonferroni correction (this is a correction for multiple testing, see also appendix A). While this correlation is somewhat conservative in the presence of spatial noise correlations (which are assumed to be relatively small due to the under-sampling existing in present fMRI's) the method is certainly valid and, like the wavelet method which will be presented here, requires no presmoothing.

Chapter 3

2D Wavelet-based brain image analysis

In this chapter, a statistical framework is presented which is later used to analyze a medical dataset. The multiresolution strategy offered by the wavelet decomposition contrasts from the random field methods, or other 'traditional' signal analysis methods, in the way the question of 'appropriate' image smoothing is approached. These methods consider the shape and/or size of the brain activation regions as known and typically apply low-pass filtering to the images, in an attempt to maximize the SNR.
The question then remains, what is the 'best' smoothing filter to use? If the activations are highly focal, then only a little smoothing would be best. Conversely, if they are diffuse, more extensive smoothing would be appropriate. The problem is further compounded by the possibility that a particular brain simulation task may elicit both types of activation patterns concurrently. Hence, a monoresolution strategy for these kind of applications is likely to be suboptimal. In an attempt to pick the 'optimal' filter for each possible activation pattern, a proposal [11] has been made to apply sequentially a set of spatially invariant and isotropic Gaussian low-pass filters with successively larger kernel widths, and then extend the search for activations over the 3D location space as well as the 1D filter scale space. However, the decomposition into a set of low-pass-filtered images is both redundant and nonorthogonal. Consequently, the number of statistical tests required to locate activations is increased, and a higher detection threshold must be selected to protect the significance level, incurring a loss of detection sensitivity [7]. In contrast to traditional, monoresolution signal detection techniques, wavelet detection is spatially adaptive and thus is able to deal in a direct and straightforward manner with signals that may have spatially heterogeneous smoothness properties, as well as a finite number of discontinuities. Hence, the question regarding the 'best' smoothing need not to be answered beforehand. Answering that question is an integral part of the wavelet-based multiresolution strategy, and the answer may be complex in that different amounts of smoothing

in different spatial neighborhoods may be required. It has been shown [3] that if nothing is known regarding the smoothness (or lack thereof) of a signal, wavelets constitute a 'near-universal' orthogonal basis.
Before the wavelet-based implementation is explained in further detail, a general introduction into wavelets and multiresolution will be presented.

3.1 Multiresolution analysis and wavelet decomposition

Wavelet decomposition and multiresolution analysis can be applied to any finite dimensional data set. For simplicity reasons, the 1D, 2D and 3D case will be represented here in this particular order (higher dimensions are simply generalizations of the 1D case).

3.1.1 The 1D wavelet transform

For simplicity, the 1D wavelet transform will be presented first.
An orthogonal wavelet transform is characterized by two functions of a continuous variable x:

1. The *scaling* function $\phi(x)$
2. The associated *mother wavelet* $\psi(x) = \sqrt{2}\sum_{k\in Z} g(k)\phi(2x-k)$

where $g(k)$ is a suitable weighting sequence. The scaling function ϕ is the solution of a two-scale equation

$$\phi(x) = \sqrt{2}\sum_{k\in Z} h(k)\phi(2x-k)$$

The sequence $h(k)$ is called the *refinement* filter. The wavelet basis functions are constructed by dyadic dilation (index j) and translation (index k) of the scaling function and mother wavelet,

$$\phi_{j,k} = 2^{-j/2}\phi(x/2^j - k)$$

$$\psi_{j,k} = 2^{-j/2}\psi(x/2^j - k)$$

It is possible to select ϕ and ψ such that $\{\psi_{j,k}\}_{(j,k)\in Z^2}$ constitutes an orthonormal basis of L_2, the space of finite energy functions. This orthogonality permits the wavelet coefficients $d_j(k)$ and approximation coefficients $c_j(k)$ of any function $f(x) \in L_2$ to be obtained as inner products with corresponding basis functions

$$d_j(k) = \langle f, \psi_{j,k}\rangle, \quad c_j(k) = \langle f, \phi_{j,k}\rangle$$

where $\langle f, g \rangle$ is the conventional L_2-inner product ($\langle f, g \rangle = \int f(x)g(x)\mathrm{d}x$).
In practice, the decomposition is carried out over a limited number of scales J. The wavelet synthesis formula with a depth J is in that case given by

$$f(x) = \sum_{j=1}^{J} \sum_{k \in Z} d_j(k)\psi_{j,k} + \sum_{k \in Z} c_J(k)\phi_{J,k}$$

Although the synthesis and analysis formulas usually are given for continuous signals, equivalent expressions also exist for a purely discrete framework. In the discrete formulation these formulas can be rewritten in a following matrix form:

$$\mathrm{f} = \mathrm{W}^T \mathrm{d}$$

$$\mathrm{d} = \mathrm{W}\mathrm{f}$$

where $\mathrm{f} = (\ldots, \mathrm{f}(k), \ldots)$ is the signal (or image) vector, W is the orthogonal wavelet transformation matrix, and $\mathrm{d} = (\mathrm{d}_1(k), \ldots, \mathrm{d}_J(k), \mathrm{c}_J(k))$ is the wavelet coefficient vector. Therefore, the wavelet transform

$$\mathrm{d} = \mathrm{W}\mathrm{f}$$

is an orthonormal transformation of the signal vector f. Instead of defining the matrix W explicitly, it can also be described by the Mallat algorithm. This algorithm starts with a multiresolution analysis with scaling function ϕ which generates an orthonormal basis of L^2. The functions

$$\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k), \quad j, k \in \mathrm{Z}$$

span the resolution subspaces V_j:

$$V_j = \mathrm{span}\{\phi_{j,k} : k \in \mathrm{Z}\}.$$

A basic wavelet ψ exists which generates the wavelet spaces

$$W_j = \mathrm{span}\{\psi_{j,k} : k \in \mathrm{Z}\},$$

where $\psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k)$, such that

$$V_{j-1} = V_j \oplus W_j.$$

The Mallat decomposition starts from a data sequence $c^0 \in l^2(\mathrm{Z})$ which is decomposed into a lowest resolution signal c^L and detail signals $d^1, d^2, \ldots, d^J$ by the recursion

$$c^j = \mathbf{H}c^{j-1}$$

$$d^j = \mathbf{G}c^{j-1}$$

where

$$(\mathbf{H}a)_k = \sum_n h_{n-2k} a_n$$

$$(\mathbf{G}a)_k = \sum_n g_{n-2k} a_n$$

The coefficients h_n and g_n are defined in terms of the scaling function:

$$h_n = \frac{1}{\sqrt{2}} \int \phi(\frac{x}{2}) \phi(x-n)\, dx$$

$$g_n = \frac{1}{\sqrt{2}} \int \psi(\frac{x}{2}) \phi(x-n)\, dx = (-1)^n h_{1-n}$$

By defining a function f corresponding to c^0 :

$$f = \sum_n c_n^0 \phi_{0,n}$$

the Mallat decomposition results in an expansion

$$f = Q_1 f + Q_2 f + \ldots + Q_L f + P_L f,$$

where

$$P_j f = \sum_k c_k^j \phi_{j,k}$$

$$Q_j f = \sum_k d_k^j \psi_{j,k}$$

are projections on subspaces V_j and W_j, respectively.
The Mallat reconstruction algorithm is recursively described by

$$c^{j-1} = \mathbf{H}^* c^j + \mathbf{G}^* d^j$$

where

$$(\mathbf{H}^* a)_k = \sum_n h_{k-2n} a_n$$

$$(\mathbf{G}^* a)_k = \sum_n g_{k-2n} a_n$$

The original signal c^0 is then obtained through:

$$c^0 = \sum_{j=1}^{L} (\mathbf{H}^*)^{j-1} \mathbf{G}^* d^j + (\mathbf{H}^*)^L c^L$$

3.1.2 The 2D wavelet transform

The wavelet decomposition

$$(d_j(k) = \langle f, \psi_{j,k} \rangle, \quad c_j(k) = \langle f, \phi_{j,k} \rangle)$$

can be extended to two dimensions (2D) by using tensor product basis functions. This is achieved by successively applying the 1D algorithm along the separate dimensions of the data.
The effect of one iteration is illustrated in Figure 3.1, for the 2D case. In this way, $2^q = 2^2$ different types of basis functions in $q = 2$ dimensions are generated. The corresponding 2D separable scaling functions with x= (x_1, x_2) are given by

$$\phi_{j,\mathbf{k}}(\mathrm{x}) = \prod_{i=1}^{2} \phi_{j,k_i}(x_i)$$

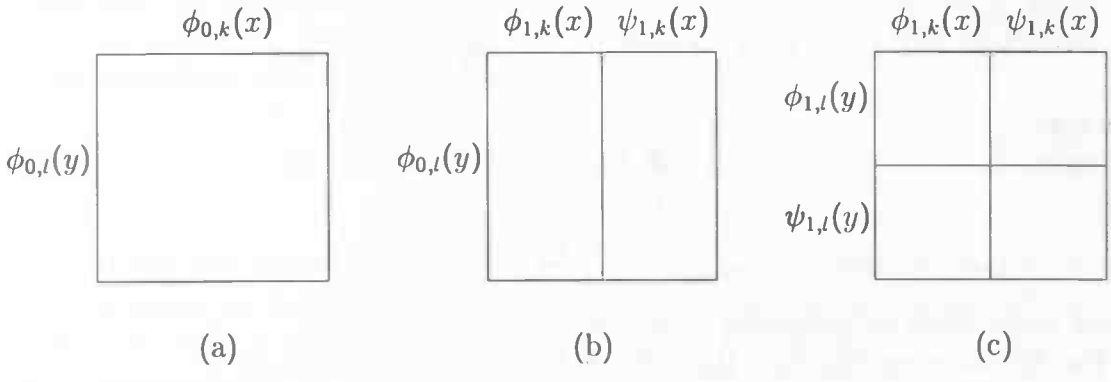

Figure 3.1: A separable wavelet transform in 2D

with vector integer index $\mathrm{k} = (k_1, \ldots, k_q)$.
The mixed tensor product wavelets are

$$w^0_{j,k}(x) = \phi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)$$
$$w^1_{j,k}(x) = \phi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)$$
$$w^2_{j,k}(x) = \psi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)$$
$$w^3_{j,k}(x) = \psi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)$$

Since ϕ is low pass and ψ is high pass, mixed tensor product wavelets tend to have a preferential spatial orientation along one (or several if $q > 2$) of the spatial directions, and m assumes the role of a spatial indicator. In this 2D case, $w_{j,k}^m$ $(m = 1, 2, 3)$ corresponds to wavelets along the vertical, horizontal and diagonal directions.

3.1.3 The 3D wavelet transform

The wavelet decomposition can be extended to three dimensions (3D) by using tensor product basis functions as well. This is achieved by successively applying the 1D algorithm along the 3 separate dimensions of the data.
The effect of two iterations is illustrated in Figure 3.2 for the 3D case. In this way, $2^3 = 8$ different types of basis functions in 3 dimensions are generated.

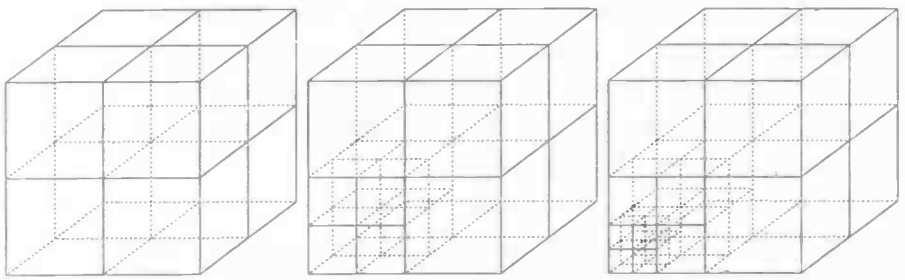

Figure 3.2: The 3 iterations in a 3-level 3D decomposition process

The corresponding 3D separable scaling functions with x= $(x_1, \ldots, x_q)$ are given by

$$
\begin{aligned}
w_{j,k}^0(x) &= \phi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)\phi_{j,k_3}(x_3) \\
w_{j,k}^1(x) &= \phi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)\psi_{j,k_3}(x_3) \\
w_{j,k}^2(x) &= \phi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)\phi_{j,k_3}(x_3) \\
w_{j,k}^3(x) &= \phi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)\psi_{j,k_3}(x_3) \\
w_{j,k}^4(x) &= \psi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)\phi_{j,k_3}(x_3) \\
w_{j,k}^5(x) &= \psi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)\psi_{j,k_3}(x_3) \\
w_{j,k}^6(x) &= \psi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)\phi_{j,k_3}(x_3) \\
w_{j,k}^7(x) &= \psi_{j,k_1}(x_1)\psi_{j,k_2}(x_2)\psi_{j,k_3}(x_3)
\end{aligned}
$$

In the 3D case, $w_{j,k}^m$ (m=0, ..., 7) correspond to wavelets which are located in all other 7 cubes as is illustrated in Figure 3.2.

3.2 Statistical framework for wavelet-based signal detection

To implement the wavelet-based detection procedure in the context of a block-design in brain activation studies (see section 2.3), the average difference image

$$\hat{f}(n) = \frac{1}{N}\sum_{i=1}^{N} g_i(n) \tag{3.1}$$

is subjected to a multiresolution decomposition according to

$$\hat{c}_j(k) = \langle \hat{f}, \phi_{j,k} \rangle \tag{3.2}$$

and

$$\hat{d}_j^m(k) = \langle \hat{f}, w_{j,k}^m \rangle \tag{3.3}$$

where $k = (k_1, k_2, \ldots, k_q)$ and where $\langle f, g \rangle = \int f(x)g(x)\mathrm{d}x$ is the conventional L_2 inner product.
In practice, the decomposition presented here is only carried out for a finite number of scales J. With $c_j(k)$ and $d_j(k)$ defined as in (3.2) and (3.3), the wavelet transform with depth J is given by

$$\hat{f}(n) = \sum_{j=1}^{J}\sum_{k} \hat{d}_j^m(k)\psi_{j,k} + \sum_{k} \hat{c}_J(k)\phi_{J,k} \tag{3.4}$$

Under the null hypothesis $g_i(n) = e_i(n)$, and $w_{j,k}^m$ perform orthonormal linear transformations on the means of N *iid* Gaussian variates with variance σ^2, resulting from (2.4) in the distribution of $\hat{d}_j^m(k)$ as *iid* $\mathcal{N}(0, \sigma^2/N)$.
Hence, standardizing the wavelet coefficients by the standard error

$$\sigma_N = \sigma/\sqrt{N} \tag{3.5}$$

with σ obtained from (2.6), yields for each of the m directional channels at resolution level j

$$\hat{d}_j^m(k)/\sigma_N \sim iid \;\; \mathcal{N}(0,1) \tag{3.6}$$

and

$$(\hat{d}_j^m(k)/\sigma_N)^2 \sim iid \;\; \chi_1^2 \tag{3.7}$$

with χ_1^2 being the chi-square distribution with 1 degree of freedom (DOF). Properties of (3.5) and (3.6), in conjunction with the orthogonality of the decomposition permit a two-stage approach to the estimation of f, which reduces the overall number of statistical tests that need to be performed.
The first stage addresses the question as to whether there is significant signal

power in any of the $(2^q-1)J$ direction-oriented resolution channels. The approximation coefficients $\hat{c}_J$ represent the extreme low-pass and DC information in the images and are routinely left unprocessed for inclusion in the subsequent signal estimation by the inverse wavelet transform.
In the second stage, only channels with significant signal power are further examined to determine the spatial location of the signal. Hence, based on (3.6), the sum of the squared, standardized coefficients in each channel is under the hypothesis $H_0 : f(n) = 0$ a chi-square variate with DOF equal to the number of summation terms. This provides the rationale for the first-stage test procedure, which jointly tests in each resolution channel the significance of the coefficients. Channels where H_0 is accepted are discarded as carrying only noise, yielding a reduced coefficient set:

$$\tilde{d}_j^m(k) = \begin{cases} \hat{d}_j^m(k), & \text{for } j = j' \text{ and } m = m', \quad \text{if } \sum_{k\in \mathrm{IC}_j}[\hat{d}_{j'}^{m'}(k)]^2 > \theta_j \\ 0 & \text{otherwise }, j = 1,\ldots,J, m = 1,\ldots,2^q-1 \end{cases} \tag{3.8}$$

where IC_j is the number of intracranial pixels at level j. The function

$$\theta_j = \sigma_N^2 \chi^2{}_{n_j^m;\alpha} \tag{3.9}$$

is the threshold at resolution level j obtained from the $(1-\alpha)$ probability cutoff of the chi-square distribution with DOF equal to the number of IC wavelet coefficients at level $j, n_j^m = \#IC_j$.
The notation $\chi^2_{a;b}$ is used to denote the chi-square distribution with DOF $a \in \mathbb{N}$ and a certain level of confidence b. The value b determines the cutoff value z_b, i.e. $\mathrm{P}(Z > z_b) = b$. So if a 95% certainty is wanted, the user has to choose value 0.05 for parameter b since $(1-0.95) = 5\%$. If each of the $(2^q-1)J$ tests is performed at a significance level of $\alpha = p/(2^q-1)J$ (Bonferroni adjustment), the overall significance per volumetric image is maintained at the specified p-level. The idea of a Bonferroni adjustment is that if k null hypotheses are to be tested, the desired overall error rate of at most α can be guaranteed by testing each null-hypothesis at level α/k, and (equivalently) they hold simultaneously with confidence level at least $100(1-\alpha)\%$. To facilitate the follow-up testing in the second stage, the index pairs (j', m') of channels with significant power are entered into a lookup table T_α.
The second stage procedure follows from (3.6) as a two-sided z-test of only the coefficients in the channels determined to carry significant power

$$\overset{\approx}{d}{}_j^m(k) = \begin{cases} \tilde{d}_j^m(k), & \text{if } |\tilde{d}_j^m(k)| > \tau \\ 0 & \text{otherwise} \end{cases} \quad (j,m) \in T_\alpha \tag{3.10}$$

In (3.8) τ functions as a threshold, with

$$\tau = \sigma_N z_{\alpha'} \tag{3.11}$$

being the threshold for the standardized normal variate with the significance level α' (remember that $P(Z > z_{\alpha'}) = \alpha'$) adjusted for the total number of follow-up tests performed in the wavelet space, i.e. the total number of IC coefficients contained in the channels considered for follow-up testing

$$\alpha' = p \Big/ \sum_{j,m \in T_\alpha} n_j^m \,. \tag{3.12}$$

Equation (3.12) constitutes the Bonferroni adjustment for multiple testing, requiring an elevation of τ relative to the single-test threshold.
The final step in the recovery of the signal is applying the inverse discrete wavelet transform to the 'surviving' coefficients, i.e.

$$f^* = W^T \tilde{\tilde{d}} \tag{3.13}$$

Hence, the estimate f^* is obtained as the sum of wavelets with coefficients that exceeded a statistically determined noise threshold. This process can be seen as a kind of adaptive noise filtering, where the filter passband is determined by the SNR levels in the various resolution channels.

3.3 Applying 2D wavelet analysis to brain images

The analysis described in the previous section has been used for an fMRI dataset which has been acquired from the academic hospital, Groningen (AZG). From this dataset a small subset has been taken, containing 10 images from which 5 have been acquired during an active state, and the other 5 were obtained during a rest state of the subject. All images are from the same subject, and have a spatial resolution of 240×240 pixels. An example of these images is the following pair:

Figure 3.3: Left image is in active state; right image is in passive state.

Following the described procedure, the first step was to create block difference images, according to (2.1). This results in $N = 5$ images, and as expected one is hardly able to see a visual result. Therefore, a standard t-test has been performed. This t-test calculates the possibility that the images contain no signal at all. The t-test has been executed with a 95% certainty value, and for all images the result was the same: the hypothesis did not hold for any of the 5 images, so the conclusion is that with 95% certainty each difference image contains a signal which is not equal to 0.
This property is confirmed when the average pixel value of $\hat{f}$ is calculated for the images:

$$\hat{f}(n) = \frac{1}{N}\sum_{i=1}^{N} g_i(n) = 4.1569.$$

Given this value, the estimated sample variance at each pixel location n can be calculated according to (2.5). The result of this calculation is a matrix with size 240×240.

Figure 3.4: Estimated sample variance.

Recall that according to (2.5) this estimated variance has DOF equal to $N-1$; in this case it is therefore 4. Using the result from above, the approximation of σ^2 can now be calculated according to (2.6):

$$\sigma^2 = \frac{1}{n_{\text{pix}}} \sum_{n \in \text{IC}} \hat{\Sigma}^2(n).$$

The summation is done over the intracranial part of the image (the pixels within the brain region). To determine which pixels are in the brain a mask is used.

Construction of this mask is based on thresholding, and it should be noted that this mask is constructed for this specific dataset. Since the mask only has values 0 (meaning the pixel is located outside brain) and 1 (pixel is within the brain), the summation in (2.6) can be calculated by multiplying each image value $\hat{f}(i,j)$ by the mask value on_mask (i,j). Using this for the fMRI dataset, we find

$$\sigma^2 \approx 14.6806$$

Figure 3.5: IC part of the brain, calculated for this dataset specifically.

The next step (according to the described procedure) is switching to the wavelet domain. The average difference image, according to (3.1), has been subjected to a multiresolution analysis according to (3.2) for the approximation and (3.3) for the detail coefficients. By using the Matlab Wavelet Toolbox this analysis has been done with a Daubechies$_5$ 2D wavelet of level 4. This results in the pictures in figure 3.6.

Figure 3.6: Average difference image $\hat{f}$ (left) and its wavelet representation (right)

As has been stated before, under the null hypothesis $g_i(n) = e_i(n)$, and $w_{j,k}^m$ perform orthonormal linear transformations on the means of N *iid* Gaussian variates with variance σ^2, resulting from (4) in the distribution of $\hat{d}_j^m$ as *iid* $\mathcal{N}(0, \sigma^2/N)$. The standard error can be easily found given the value of σ as determined by (2.6):

$$\sigma_N = \sigma/\sqrt{N} = \sqrt{14.6806}/\sqrt{5} \approx 1.7134.$$

Adjusting the wavelet coefficients by dividing them by σ_n results in only a slightly different visual result which therefore is not shown here. According to (3.7) a threshold θ_j has to be found for each scale j. In (3.7) there is a variable n_j^m which represents the number of IC pixels at each resolution level. By using Matlab with

the sum command and the constructed on_mask matrix as parameter, an estimate for the number of IC pixels in the original scan has been calculated. The value found is 6702, and based on this value we computed estimates for the number of IC pixels at each level. The coefficients at level 1 are represented by a matrix for the horizontal, diagonal and vertical channel, and each of these matrices have size $x/2 \times y/2$ if the original scan has size $x \times y$, so the number of pixels of a level 1 detail channel matrix is $\frac{1}{4}$ times the number of pixels in the original scan. Given this information and the fact that there are 3 channels at each resolution level, the following table can be made:

Resolution level	# IC pixels	# IC pixels per channel
Original scan	6702	
Level 1	5028	1676
Level 2	1257	419
Level 3	314	105
Level 4	79	26

Given these numbers of pixels, the values of θ_j can be calculated for all resolution levels j. The value of α in (3.7) has been taken as 0.05 so that a 95% certainty level is maintained concerning the values.

Based on (3.9) the following values for θ_j have been determined by using the Matlab chi2inv routine:

Resolution level j	θ_j
4	697
3	281
2	160
1	67

When thresholding is applied (using the Wavelet Toolbox for Matlab) and at each resolution level j the appropriate value θ_j is used, a reduced coefficient dataset remains. A graphical representation of these coefficients is omitted, since

it is much more interesting to show a reconstruction in the spatial domain by using only the remaining significant coefficients in the inverse wavelet transform:

Figure 3.7: Reconstruction by using only significant coefficients (left) and final result after second stage procedure (right).

The final step in the wavelet analysis is determining the value of τ and applying the second stage procedure. Using (3.11) the value 220.3341 for τ is found, and when the coefficients are thresholded according to (3.10) the result after an inverse wavelet transform is the right image in figure 5. The lighter areas are the most active parts of the brain according to this wavelet analysis.

Chapter 4

3D Wavelet-based brain image analysis

In this chapter, a statistical framework is presented which is later used to analyze medical data. The framework is an extension of the 2D analysis which was presented in the previous chapter.

4.1 Statistical framework for wavelet-based signal detection

To implement the wavelet-based detection procedure, the average difference image

$$\hat{f}(n) = \frac{1}{N}\sum_{i=1}^{N} g_i(n) \qquad , n \in \mathbb{Z}^3 \tag{4.1}$$

is as in the 2D case subjected to a multiresolution decomposition according to (3.2) and (3.3), using a 3D wavelet basis. Because the theoretical part of the 3D wavelet-based analysis is analogous to its 2D equivalent, it will therefore not be discussed here.

4.2 Applying 3D wavelet analysis to brain images

The 3D analysis has been applied to another dataset (which has also been obtained from the AZG hospital). This dataset contained 4 times 78 volume scans. Each volume scan contains 24 slices. The scans are grouped, and there are 6 volume scans in each group. The first group consists of volume scans $1 \ldots 6$, the second group consists of volume scans $7 \ldots 12$ etc.
The data comes from an experiment with a subject performing some memory

tasks, alternating between a rest condition and an activation condition, with a total of thirteen periods. Each condition lasted 42 seconds and a total of six volume scans were made in each condition. The experiment was set up in the following way:

- 6 images: subject in rest (only a fixation point)
- 6 images: subject performing a memory task (brightness patterns)
- 6 images: subject in rest
- 6 images: subject performing an alternative memory task (character)
- 6 images: subject in rest (only a fixation point)
- 6 images: subject performing a memory task (red/green patterns)
- 6 images: subject in rest
- 6 images: subject performing an alternative memory task (character)
- 6 images: subject in rest (only a fixation point)
- 6 images: subject performing a memory task (blue/yellow patterns)
- 6 images: subject in rest
- 6 images: subject performing an alternative memory task (character)
- 6 images: subject in rest

The entire set consists of 4 subdirectories, each with volume scans organized as listed here.
For the 3D wavelet-based analysis, only one subdirectory at a time has been analyzed per experiment. Two series of scans have been taken (so each series contains six scans ,which indicates $N = 6$) and the wavelet-based analysis has been applied to determine significant changes between the two groups. For example, in one experiment scans $1 \ldots 6$ have been taken as a group without stimulus, and scans $67 \ldots 72$ as a group of scans with a stimulus.
The next issue is the choice of wavelet type. Matlab provides a lot of wavelet types, i.e. the Daubechies wavelet, and several spline wavelets. However, the statistical analysis requires the use of orthogonal wavelets. Because the entire Daubechies family of wavelets is orthogonal, these are the wavelets which have been used here.
As mentioned before, large parts of the 3D analysis are completely analogous to its 2D counterpart.
Some statistical values in the 3D case:

- N = 6
- nSlices = 24
- $\sigma^2 \approx 2039$
- The standard error, $\sigma_N = \sigma/\sqrt{N} \approx 18.43$

After a 3D mask has been created in a similar way as in the 2D case, the following properties of the dataset have been determined when a decomposition of level 3 has been performed with a Daubechies$_4$ type wavelet:

Resolution level	# IC voxels	# IC voxels per channel
Original scan	82019	
Level 1	10252	1282
Level 2	1282	160
Level 3	160	20

The next step was determining a threshold for the first stage detection part (which depends on the decomposition level):

Resolution level	Channel threshold θ_j
Level 1	868434741
Level 2	104024924
Level 3	11442742

The second stage detection was to determine a threshold for every element in the remaining reduced dataset (the dataset with the insignificant channels left away). This threshold was the value τ. For this dataset, $\tau \approx 294.4$ (this value is for a threshold in the wavelet domain). After the second stage detection procedure, the remaining dataset is transformed back. Below are some results of the 3D wavelet-based analysis. Slices $1 \ldots 6$ have been compared to slices $61 \ldots 66$. The slices with the largest activity are shown (slices $8 \ldots 24$, where slice 24 is the lowest scan in the brain, that is, the slice which is closest to the chest of the subject).

The most interesting slices are shown in a bigger format, which is interesting because they will be compared to an AFNI analysis later. The data can also be viewed from other directions (not just from above as presented here), but unfortunately the AZG dataset has only 24 slices in the z-direction. This means that the other two projection possibilities would show only 128×24 pixels.

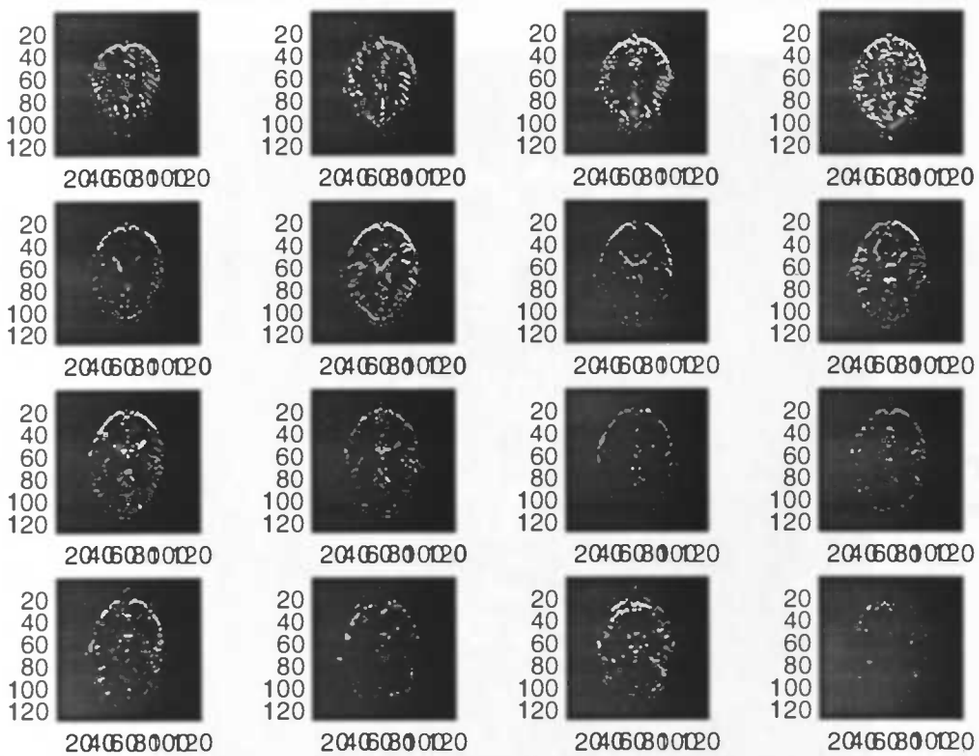

Figure 4.1: Result of 3D wavelet-based analysis of slices 8 . . . 24. Blue is the lowest intensity, yellow is the highest intensity with. The used wavelet type is Daubechies3, at level 3.

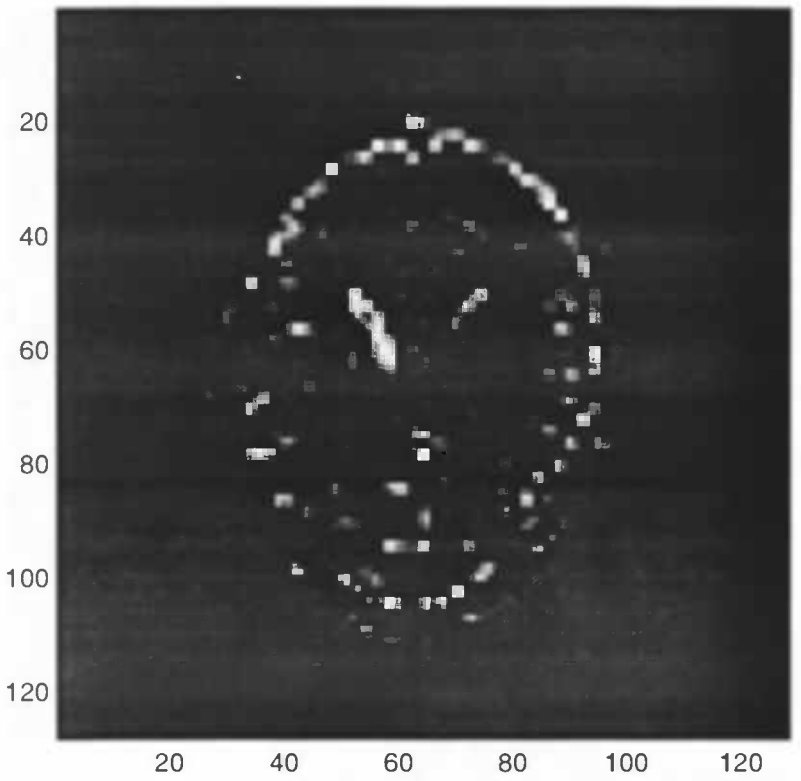

Figure 4.2: Slice 10 of a 3D wavelet analysis (level 3, Daubechies4 wavelet)

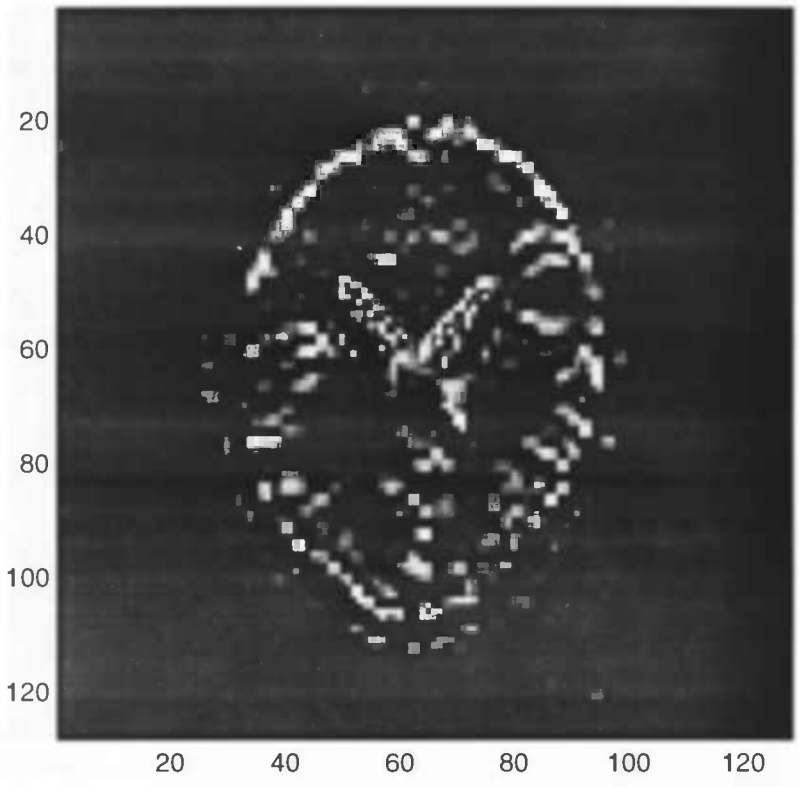

Figure 4.3: Slice 11 of a 3D wavelet analysis (level 3, Daubechies4 wavelet)

Chapter 5

Comparing wavelet analysis to conventional analysis methods

To verify the results which have been found in the previous chapter, an analysis in the spatial domain has been done using an fMRI analysis software suite called AFNI. AFNI is a program created by Robert W. Cox (`rwcox@mcw.edu`). For a detailed description of obtaining, installing and using this program, see

`http://www.biophysics.mcw.edu/BRI-MRI/afniregister.html`

In the following sections an analysis of the AZG dataset by AFNI is presented.

5.1 Realigning the dataset

Before the dataset, which is stored in a so-called Analyze format, is converted to the AFNI file format, its separate slices have to be realigned. The AZG dataset consists of separate slices, each with a spatial dimension of 128×128 pixels. Per scan of a brain volume, 24 of those slices are generated.
Slices are realigned by using the AFNI subprogram `imreg`. Imreg takes as input argument a series of 2D images and a so-called base–image. The output of `imreg` is a series of images, which have been registered with respect to the base–image of the input. Besides the output images, `imreg` also states $\Delta x, \Delta y$, and $\Delta\theta$ for each slice. These symbols stand for the translation in horizontal and vertical directions (measured in pixels), and the rotation angle, respectively.
The AZG dataset contains a very large number of slices ($4 \times 24 \times 78 = 7488$ slices), which cannot be handled entirely by `imreg`: the dataset has to be split into smaller pieces. Another problem with the dataset is the name of the files of the separate slices which, at first sight, is rather unstructured. A Pascal program has been written which generates a shell script which renames all slices to the form `rolXX-YY.ima` where `XX` denotes the scan number (01—78), and `YY` denotes the slice number (01—24). This however does not solve the previous problem of

imreg not being able to handle such a large number of slices. To solve this, a python script which moves all slices of a certain scan into a separate subdirectory was written:

```
#!/usr/bin/python
# Script to move all slices of scan-number XX to directory "XX"

import os
for i in range(78):
os.system("mkdir %02d; mv rol%02d* %02d/" % (i+1, i+1, i+1))
```

The next problem is that the files from the AZG dataset are in raw scanner file format. Since AFNI can only handle files of the Analyze format, the entire dataset has to be converted to this format.
This has been done by using medcon, a program which can convert files to a lot of other medical file formats. For further information about medcon, we refer to

http://petaxp.rug.ac.be/~nolf/

Since it is rather annoying to run medcon manually in each subdirectory of the dataset, a wrapper has been written in python:

```
#!/usr/bin/python
# Wrapper for medcon. After realigning, old files are removed.

import os
for j in range(78):
os.chdir("%02d" % (j+1))
os.system("""medcon -c "anlz" -f *.ima""")
os.system("rm *.ima")
os.chdir("..")
```

Now that there are 78 subdirectories with 24 slices each, the dataset is suitable for realigning by imreg. A wrapper in python simplifies this:

```
#!/usr/bin/python
# Script to call imreg. Base-image is the average of the slices.

import os
for i in range(78):
os.chdir("%02d" % (i+1))
os.system("~/imagers/afni98/imreg +AVER *.img")
```

```
os.system("rm *hdr")
os.system("rm *img")
os.chdir("..")
```

Along the traversal through the subdirectories some cleaning-up is accomplished by removing files which will not be used further.

5.2 Theory behind `imreg`

Registration is accomplished by minimizing the error functional

$$E(I,J) = \sum_{x,y} [I(x,y) - J(T(u,v,h)(x,y))]^2 \tag{5.1}$$

where $J(x,y)$ is the base–image, $I(x,y)$ is the image to be registered to it, $T(u,v,h)$ is the transformation with shift parameters (u,v) and rotation angle h. The minimization is then a least squares problem. The (u,v,h) that minimizes E is found. At this point, $I(x,y)$ is transformed with $T(-u,-v,-h)$ to bring it closer to $J(x,y)$ (bicubic interpolation is used for resampling I). Then the minimization is repeated – that is, simple gradient descent is used to minimize E.
Actually, this procedure is performed with a $J(x,y)$ that has been smoothed with a Gaussian filter with FWHM=4 pixels. In this way, the effects of pixels a little further away than nearest neighbors are included in the minimization, and displacements of up to 2–3 pixels can be detected.
When the FWHM=4 pixel smoothing iteration converges, it is repeated with $J(x,y)$ smoothed with FWHM=1.0 pixels. Typically, 2–4 iterations are required to converge in the first step, and 1–2 in the second if the displacements are larger that $\frac{1}{2}$ pixel. The Gaussian smoothing and the differentiation of $J(x,y)$, which is necessary in the minimization process, are both done with FFT's [2].

5.3 Putting separate slices in a 3D dataset

Now that the slices have been renamed, converted, splitted and realigned, the 24 slices of a scan can be merged to one 3D dataset. The sole reason for this is that AFNI can only perform an analysis on these 3D datasets, and not on separate slices. In the AFNI software suite, this can be done by using the `to3d` program. Since calling `to3d` with its appropriate parameters 78 times is not desirable, a python wrapper has been written for `to3d`:

```
#!/usr/bin/python
# Wrapper for to3d. Realigned slices are removed afterwards.
```

```
import os
for i in range(78):
os.chdir("%02d" % (i+1))
os.system("~/imagers/afni98/to3d \
                    -xFOV R-L -yFOV A-P  -zSLAB S-I \
                    -anat -session 3d -prefix 3d reg.????")
os.system("rm reg.00??")
os.chdir("..")
```

A short explanation of the parameters of `to3d`:

- `-anat`
  Writes information in the generated header of the 3D dataset, indicating that the scan is an anatomical MRI scan. This doesn't influence the 3D dataset, but provides the user with some background information of the scan.

- `-xFOV R-L`
  Adds information about the orientation of the scanned brain volume. The x-axis of the dataset goes from the right to the left side of the brain. The left side of the brain is near the subject's left ear, the right side of the brain is near the subject's right ear.

- `-yFOV A-P`
  Adds information about the orientation of the scanned brain volume. The y-axis of the dataset goes from the *anterior* to the *posterior* side of the brain. The anterior part of the brain is located near the eyes of a subject, the posterior is at the back of the head.

- `-zSLAB S-I`
  Adds information about the orientation of the scanned brain volume. The z-axis of the dataset goes from the *superior* to the *inferior* side of the brain. The superior side is located in the top region of the brain, the inferior side is at the side of the neck of a patient.

- `-prefix 3d`
  This parameter specifies the prefix of the name of the dataset being generated. If a dataset already exists with the same prefix name, then the old dataset will not be overwritten. This is also made visible to the user by a warning message on the screen.

- `-session 3d`
  This parameter indicates in which subdirectory the generated dataset will be written. If no subdirectory with the given name exists, it will be made on-the-fly by `to3d`.

- `reg.????`
  This last parameter specifies the input-files. In this particular case, the input files are named `reg.0001`, `reg.0002`, ..., `reg.0024` as they are the output files of the `imreg` program. Wild-cards are allowed, and have been used here in the form of a question mark, but the asterisk symbol can also be used.

When this step has been performed, the result is a 3D dataset. This 3D dataset can be viewed with the main AFNI program, which is named `afni`. The `afni` program expects a parameter which represents the directory in which the 3D dataset is located. Therefore, if the dataset is located in a subdirectory called `3d`, then afni should be started with the command `afni 3d/`.
When `afni` is started, the user can select different views on the 3D dataset (these are a sagittal view, a coronal view and an axial view). For more information, please see [2]. For an overview of all options which can be passed to `to3d` , please see [17].

5.4 Statistical analysis in AFNI

5.4.1 Introduction

The main reason for using AFNI is being able to perform a statistical analysis based on a number of functional MRI scans. In practice this means that

1. First an anatomical dataset is made
2. After that a number of functional datasets are displayed on the anatomical set.

In AFNI it is necessary to map a functional dataset to an anatomical dataset. The user is forced to do this, because `afni` will not start if there is not an anatomical 3D dataset in the directory specified by the user.

5.4.2 Purpose of 3dANOVA

Statistical analysis can be performed with the program `3dANOVA` of the AFNI software suite. 3dANOVA (*3D ANalysis Of VAriance*) was designed to perform a so-called *voxel-per-voxel 1-factor* analysis. The user specifies, using a command-line interface, which AFNI 3D data sets are to be used in the analysis, and at which *factor level* they are. An explanation of what a factor level is, will be given in the next section.
Some statistical properties and models that are supported by `3dANOVA` are performing an F-test to compare 2 factor levels, giving an estimation of the difference between 2 factor levels, estimating the averages of individual factor levels, and

calculating an estimation of contrasts. The result of an analysis can be saved either in different AFNI *sub-brick* datasets or as a single *bucket* type dataset.

5.4.3 Theory of 3dANOVA

Analysis of variance is a statistical technique to study the relation between a dependent variable with regard to one or more independent variables. ANOVA does not make assumptions regarding the functional relation between these two kinds of variables, and ANOVA also does not assume that the independent variables are quantitative, and they are therefore allowed to be qualitative. In ANOVA the independent variables are called *factors*. *3DANOVA* assumes that there is only one factor. To perform a 2-factor analysis, another AFNI program exists with the name `3DANOVA2`. In AFNI there is as well a tool for 3-factor statistical analysis; this can be done with the `3DANOVA3` program.
Calculating an ANOVA for fMRI datasets is complicated because of the enormous amount of ANOVA's that have to be calculated: one per voxel. In practice this comes down to millions of ANOVA's having to be calculated for a real-world dataset. A sequential computation can therefore take quite a while.

The ANOVA model

In this model one assumes that there is only one factor. Another assumption which applies to this model is that there are studied r different *levels* (also called *treatments*), and further that at factor level i there are n_i observations ($i = 1, \ldots, r$).
In a schematic representation:

1	2	...	r	$\leftarrow$	factor levels
Y_{11}	Y_{21}	...	Y_{r1}	$\leftarrow$	data
Y_{12}	Y_{22}	...	Y_{r2}		
$\vdots$	$\vdots$	$\ddots$	$\vdots$		
Y_{1n_1}	Y_{2n_2}	...	Y_{rn_r}		

Each Y_{ij} represents an observation of the same voxel in different AFNI 3D datasets. The total number of observations is represented by n_t, with

$$n_t = \sum_{i=1}^{r} n_i \tag{5.2}$$

Then the *fixed effects* ANOVA model is described by

$$Y_{ij} = \mu_i + \epsilon_{ij}, \qquad (i = 1, \ldots, r; \quad j = 1, \ldots, n_i) \tag{5.3}$$

with

Y_{ij} = the j–th observation at factor level i,
μ_i = average response at factor level i,
ϵ_{ij} = random variable, statistically distributed following a $\mathcal{N}(0, \sigma^2)$ distribution.
The total number of observations at factor level i is denoted by $Y_{i.}$:

$$Y_{i.} = \sum_{j=1}^{n_i} Y_{ij} \ , \tag{5.4}$$

and the total of all observations is denoted by $Y_{..}$:

$$Y_{..} = \sum_{i=1}^{r} \sum_{j=1}^{n_i} Y_{ij} \ . \tag{5.5}$$

The *sample mean* at factor level i is denoted by $\bar{Y}_{i.}$:

$$\bar{Y}_{i.} = \frac{\sum_{j=1}^{n_i} Y_{ij}}{n_i} = \frac{Y_{i.}}{n_i} \tag{5.6}$$

and the *overall* average (denoted by $\bar{Y}_{..}$):

$$\bar{Y}_{..} = \frac{\sum_{i=1}^{r} \sum_{j=1}^{n_i} Y_{ij}}{n_t} = \frac{Y_{..}}{n_t} \ . \tag{5.7}$$

The deviation of an individual observation relative to the overall average can be expressed by:

$$\underbrace{Y_{ij} - \bar{Y}_{..}}_{\substack{\text{total deviation} \\ \text{w.r.t. overall} \\ \text{average}}} = \underbrace{Y_{ij} - \bar{Y}_{i.}}_{\substack{\text{deviation w.r.t.} \\ \text{factor level} \\ \text{average}}} + \underbrace{Y_{i.} - \bar{Y}_{..}}_{\substack{\text{deviation of factor} \\ \text{level average w.r.t.} \\ \text{overall average}}} \tag{5.8}$$

If the square of both sides is taken, and a summation is made over all observations and all factor levels, this results in:

$$\underbrace{\sum_{i=1}^{r} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{..})^2}_{\text{SSTO}} = \underbrace{\sum_{i=1}^{r} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i.})^2}_{\text{SSE}} + \underbrace{\sum_{i=1}^{r} n_i (\bar{Y}_{i.} - \bar{Y}_{..})^2}_{\text{SSTR}} \tag{5.9}$$

Here the following abbreviations are introduced:

- SSTO: total sum of squares
- SSTR: treatment sum of squares (between groups)

- SSE: error sum of squares (within groups)

The averages of the squares can be obtained by dividing the sum of squares by the corresponding number of degrees of freedom (DOF's):

$$\text{MSTR} : \frac{\text{SSTR}}{r-1} \tag{5.10}$$

$$\text{MSE} : \frac{\text{SSE}}{n_t - r} \tag{5.11}$$

(MS = Mean Square). All the information is combined into an ANOVA table:

Source of variation	SS	df	MS
Between treatments	$SSTR$	$r-1$	$MSTR = \frac{SSTR}{r-1}$
Within treatments	SSE	$n_t - r$	$MSE = \frac{SSE}{n_t - r}$
Total	$SSTO$	$n_t - 1$	

(5.12)

df = degrees of freedom, MS =Mean Square.

Equality test for factor level averages

To test the null-hypothesis (*all factor levels are equal*):

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_r \tag{5.13}$$

against the alternative hypothesis:

$$H_a : \text{not all } \mu_i \text{ are equal} \tag{5.14}$$

the next test statistic is used:

$$F^* = \frac{MSTR}{MSE} \tag{5.15}$$

If F^* is large, then the outcome tends to H_a, and if F^* is small, the outcome tends to the null-hypothesis H_0. It can be proved that if H_0 holds, then F^* follows a $F(r-1, n_t - r)$ distribution where F denotes the F–distribution with $(r-1)$ and $(n_t - r)$ degrees of freedom (see [17] and [12]). Let F_α be determined by

$$\text{P}(F > F_\alpha) = \alpha$$

The following rule of decision is used:

$$\text{If } F^* \leq F_\alpha, \text{conclude } H_0 \ ,$$

$$\text{If } F^* > F_\alpha, \text{conclude } H_a \ .$$

Once it has been determined that there exists a treatment effect (that is, not all factor levels are equal), there has to be examined which treatments are responsible for rejecting the null-hypothesis.

Estimating the difference between two factor levels

If it has been determined that there exists a treatment effect, then a logical next step is to take two different treatments pairwise, and to look for significant differences between these two. The difference D_{ij} between 2-factor averages μ_i and μ_j ($D_{ij} = \mu_i - \mu_j$), can be estimated by

$$\hat{D}_{ij} = \bar{Y}_{i.} - \bar{Y}_{j.} \tag{5.16}$$

The estimated difference of variance $\hat{D}_{ij}$ is given by

$$s^2(\hat{D}_{ij}) = MSE(\frac{1}{n_i} + \frac{1}{n_j}) \tag{5.17}$$

This means that $\frac{\hat{D}_{ij} - D_{ij}}{s(\hat{D}_{ij})}$ follows a $t(n_t - r)$ distribution, and that the $(1 - \alpha)$ significance interval for D_{ij} is given by

$$\hat{D}_{ij} \pm t(1 - \frac{\alpha}{2}; n_t - r)s(\hat{D}_{ij}) \tag{5.18}$$

Therefore, if $t^* = \frac{\bar{Y}_i}{s(\bar{Y}_{i.})}$ has a large absolute value, then this suggests that the ith and the jth factor averages are indeed different.

5.4.4 Using 3dANOVA on the AZG dataset

The most comfortable way of using `3DANOVA` is by first writing a Unix shell script. An example of such a script is:

```
3dANOVA -levels 2     \
-dset 1 p1/01_3d+orig \
-dset 1 p1/02_3d+orig \
-dset 1 p1/03_3d+orig \

{ Etcetera ... }
```

```
-dset 1 p4/04_3d+orig \
-dset 1 p4/05_3d+orig \
-dset 1 p4/06_3d+orig \

{ Etcetera ... }

-dset 2 p1/73_3d+orig \
-dset 2 p1/74_3d+orig \
-dset 2 p1/75_3d+orig \

{ Etcetera ... }

-dset 2 p4/76_3d+orig \
-dset 2 p4/77_3d+orig \
-dset 2 p4/78_3d+orig \
-voxel 1              \
-diff 1 2 D1-D2       \
-ftr Ftest
```

In fact, this script is only one command, 3DANOVA with a lot of parameters. The purpose of the first line is to start the 3DANOVA program, all other lines are parameters for this program.
The second line of the script, ''-levels 2'' instructs 3DANOVA to compare two different types of datasets. In this particular case, these are on the one hand the scans in which the subject was given a stimulus during the making of the scan, and on the other hand the scans where such a stimulus was not present.
All the lines which start with ''-dset'' instruct 3DANOVA to insert a 3D dataset in a certain database of scans which are to be grouped together. In this particular case, there are two such groups, which are designated '1' and '2'. These groups are also called an *on*–set and an *off*–set. It is of no importance in which group ('1' or '2') an on–set or off–set is placed. The statistical analysis is independent of this. Note that in this case there are 4 subdirectories called p1, p2, p3 and p4 because the experiment has been repeated 4 times.
The line ''-voxel 1'' instructs ''3DANOVA'' to print the value of voxel 1 to the screen while computing the analysis, and merely functions as a progress indicator to the user.
The line ''-diff 1 2 D1-D2'' instructs 3DANOVA to calculate a *t*–test on the specified sets (1 and 2). The result is stored in two files, D1-D2+tlrc.HEAD and D1-d2+tlrc.BRIK where the .HEAD file is a header file, and the .BRIK file is the data file. 3DANOVA calculates the difference between 2-factor level averages at level 1 and 2, and the corresponding t–statistics for each voxel. The result is a *fitt*–type dataset (Functional Integrity with T-Test). The prefix of the dataset which is to be generated is also specified by the user (in this case it is D1-D2).

The line ''-ftr Ftest'' instructs 3DANOVA to calculate an F–test to test for equality of factor level averages. The statistic is applied to every voxel; the result is written to two files, again a .BRIK–file and a .HEAD–file, with prefix Ftest as has been specified here. The result is a *fift* dataset (Functional Integrity with F-Test) with two sub-bricks. The first sub-brick (I) contains a $\sqrt{MSTR}$ value for each voxel, and the second sub-brick contains the corresponding F–statistics. These are calculated by:

$$F^* = \frac{MSTR}{MSE} \tag{5.19}$$

5.4.5 Automatic generator for 3dANOVA scripts

The data comes from an experiment with a subject performing some memory tasks, alternating between a rest condition and an activation condition, with a total of thirteen periods. Each condition lasted 42 seconds and a total of six volume scans were made in each condition. The experiment was set up in the following way exactly the same way as has been described in section 4.2.

The volume scans consist of 24 slices with a resolution of 128×128 pixels and a voxel-size of (2, 2, 4) mm. The Z-area covered almost the entire brain. The scans were not taken parallel to the x-y plane. There was a little rotation around the x-axis.

The 3D datasets have been numbered in the order as they are presented here, from 01 to 78. Because it is quite annoying to write an ANOVA shell script every time that an experiment is wanted on another piece of the dataset, a python script has been written to automatically generate such a script. This python script takes requires 2 parameters: the number of the first image which is in the *on*–set, and the number of the first image in the *off*–set:

```
#!/opt/hppd/bin/python
#Automatically generate an anovascript called 'ano'

import os
from sys import *

if len(argv)==3:
  if abs((float(argv[1]) - float(argv[2]))) >5:
    os.system("""echo "3dANOVA -levels 2 \\\\" > ano""")
    for j in range(4):
      for i in range(6):
        os.system("""echo "-dset 1 p%d/%02d_3d+orig \\\\" \
                  >> ano""" % ((j+1),(float(argv[1])+i)))
        os.system("""echo "-dset 2 p%d/%02d_3d+orig \\\\" \
                  >> ano""" % ((j+1),(float(argv[2])+i)))
    os.system("""echo "-voxel 1 -diff 1 2 D1-D2 \
```

```
                -ftr Ftest" >> ano""")
    os.system("chmod 700 ano")
  else:
    print "Arguments do not make sense."
else:
  print "Incorrect number of parameters (2 expected)."
```

5.4.6 Results

With the method which was described in the previous subsections, a number of images have been created with AFNI. The images contain an anatomical background, and are in gray-scale format. The functional data from the ANOVA analysis is shown in a colored palette.
When the functional data are examined, it becomes clear that the activity in the brain is limited almost always the 10th and 11th slice. All other slices contain no significant functional values (they only show activity if the significance-level is far beneath 95%).
Another interesting fact is that in some kind of experiments there is nearly no activity measurable, and that in two other experimentally acquired scans there is sometimes a lot of activity present. An example of two different series of scans where no activity is measurable is a division of slices 7 through 12 in set 1, and slices 13 through 18 in set 2. If an ANOVA script is made with the gens 7 13 command, there are no colored pixels in any slice. Note that slices 7...12 are the first set of a subject stimulated with brightness patterns, and that slices 13...18 is the first set of slices where the subject is in a rest state.
The images which are displayed here are the difference test and F–test of a scan series 01...06 and 61...66. The first series of scans concerns a subject who is focused on a fixation point, and the second series of scans concerns a subject in a complete rest state.

5.5 Comparison between AFNI and wavelet approach

To make a comparison between the results of AFNI and the wavelet based approach for signal detection, a close-up of some scans has been generated.

The images which have been created using AFNI, tend to contain less structure with regard to regions with larger activations. Activated parts of the brain are at the same physical location as has been determined by the wavelet-based approach, but consists much more of individual pixels in the view-plane.
However, by using more images (for example, when $N = 24$ is used instead of

$N = 6$), the results of the AFNI analysis tend to converge to the results that are obtained when wavelets are used. The areas with activation are smoother.

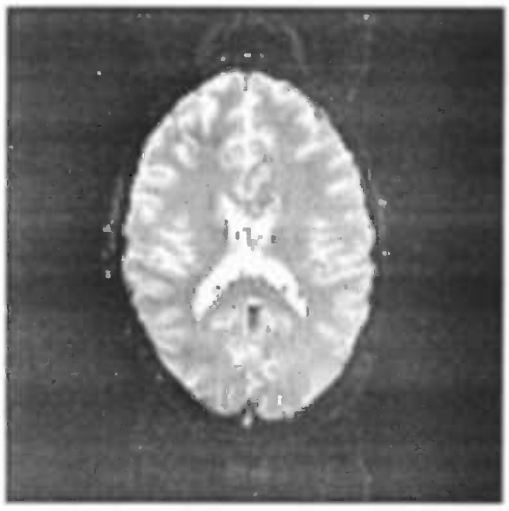

Figure 5.1: F-test ($\alpha = 95\%$) at slice 10, $N = 24$

Figure 5.2: F-test ($\alpha = 95\%$) at slice 11, $N = 24$

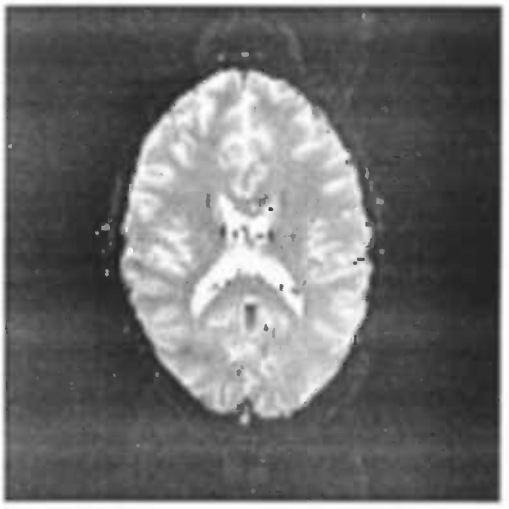

Figure 5.3: t–test ($\alpha = 95\%$) at slice 10, $N = 24$

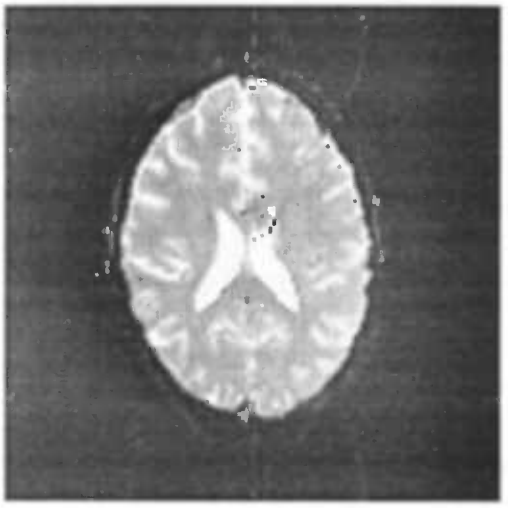

Figure 5.4: t–test ($\alpha = 95\%$) at slice 11, $N = 24$

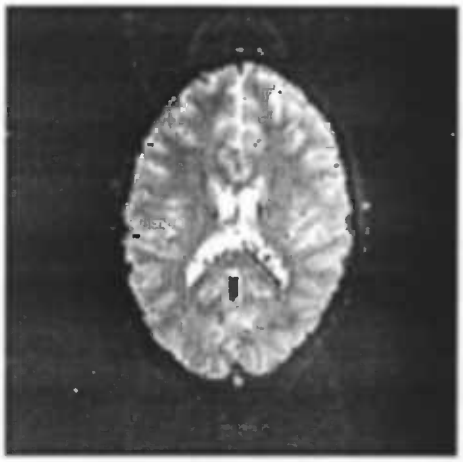

Figure 5.5: AFNI t-test ($\alpha = 95\%$) at slice 10, $N = 6$

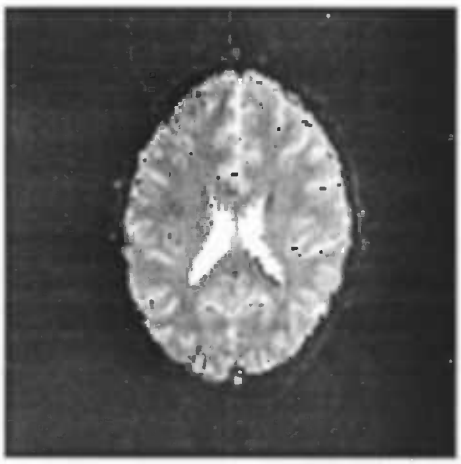

Figure 5.6: AFNI t-test ($\alpha = 95\%$) at slice 11, $N = 6$

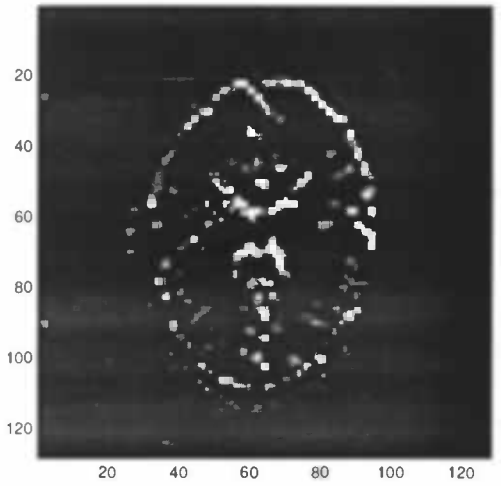

Figure 5.7: Wavelet-based test, 95 %, db3 level 3 slice 10

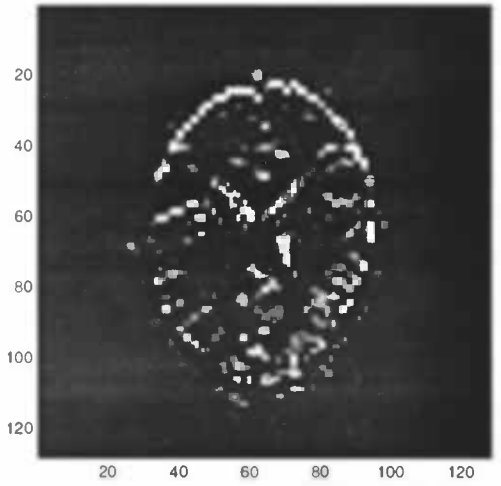

Figure 5.8: Wavelet-based test, 95 %, db3 level 3 slice 11

Chapter 6

Conclusion and recommendations for further research

The results of the wavelet-based detection algorithm are promising. If fewer images are used as in a conventional statistical approach, the same results can be obtained easily. This indicates that the wavelet detection algorithm is capable of handling noise better than a conventional approach. Wavelets with a larger size (for example the Daubechies 3, 4 and 5) tend to generate better results, because the resulting images show less very small areas with high activity (e.g. single pixels). However, a disadvantage of using larger wavelets is that the wavelet transform will take longer, and that one can not use very high levels in the decomposition on low-resolution scans.
A disadvantage of using the wavelet-based statistical approach is that it requires far more processor time and memory than a conventional approach. An analysis with AFNI takes only two minutes, and an analysis with the wavelet approach can take up to 15 minutes if a large wavelet is taken, or if a large level is used. The main reason for this, is that the 3D wavelet transform has to be applied before running the statistical tests, and after that, an inverse 3D wavelet transform has to be done to obtain results which are in the spatial domain. The implementation which has been created and used in this research consists of Matlab code only, and therefore things can be speeded up by writing an implementation in C.
A few recommendations for further research:

- Other directions
  The data sets which have been used in this research, contain volume scans of size $128 \times 128 \times 24$ pixels, and therefore only one view direction was suitable. However, volume scans of $128 \times 128 \times 128$ for example are far more suitable for an arbitrary projection.

- 3D volume renderer
  The data can also be used as input for a 3D volume renderer. An advantage

of such a view on the data is that the user can get a better idea of the physical location of the active parts of the volume scans.

- Implementation in C or Fortran
  To speed up the process, the entire implementation could be rewritten in C. Another option is to write the computationally involved parts of the implementation in CMAX, the Matlab C extension.

- Use of other wavelets
  The statistical analysis requires the use of orthogonal wavelets, which is the main reason why the Daubechies family of wavelets has been used. The use of non-orthogonal wavelets therefore requires changing the statistical framework as well.

Appendix A

Statistics for testing experimental results

This appendix is a summary of some elementary statistical models which are widely-used in experimental setups. For a more comprehensive explanation, with additional examples, see [1].

A.1 Distributions Derived from Normal Distribution

A.1.1 χ^2, t and F Distributions

Definition: If Z is a normal random variable, then the distribution $U = Z^2$ is called the χ^2 distribution with 1 degree of freedom; also denoted by χ^2_1. Please note that if $X \sim N(\mu, \sigma^2)$ then $(X-\mu)/\sigma \sim N(0,1)$, and therefore $[(X-\mu)/\sigma]^2 \sim \chi^2_1$.

Definition: If $U_1, U_2, \ldots, U_n$ are independent chi-square random variables with 1 degree of freedom, the distribution of $V = U_1 + U_2 + \cdots U_n$ is called the chi-square distribution with n degrees of freedom; also denoted by χ^2_n. It can be shown that $E(V) = n$ and $Var(V) = 2n$. Given the definition of this distribution, it can be shown that if $U \sim \chi^2_n$ and $V \sim \chi^2_m$ then $U + V \sim \chi^2_{m+n}$.

Definition: If $Z \sim N(0,1)$ and $U \sim \chi^2_n$ and Z and U are independent, then the distribution of $Z/\sqrt{U/n}$ is called the t distribution with n degrees of freedom, denoted by t_n.

A.1.2 The sample Mean and the Sample Variance

Let $X_1, X_2, \ldots X_n$ be independent $N(\mu, \sigma^2)$ random variables; they will be referred to as a **sample** from a normal distribution. In this section we will find the joint and marginal distributions of

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

These are called the sample mean and the sample variance, respectively. Please note that since $\bar{X}$ is a linear combination of independent normal random variables, it is normally distributed with

$$\mathrm{E}(X) = \mu$$

$$\mathrm{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

Theorem A: The random variable $\bar{X}$ and the vector of random variables $(X_1 - \bar{X}, X_2 - \bar{X}, \ldots, X_n - \bar{X})$ are independent.

The proof of this theorem (as any proof of any theorem in this summary) can be found in [12].

Corollary A: $\bar{X}$ and S^2 are independently distributed.

Theorem B: The distribution of $\frac{(n-1)S^2}{\sigma^2}$ is the χ^2_{n-1} distribution.

Corollary B: Let $\bar{X}$ and S^2 be as given at the beginning of this section. Then

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}$$

A.2 Comparing Two Samples

A.2.1 Two independent Samples

In many experiments, the results from an experiment differ from time to time. This occurs even if the circumstances are identical, e.g. in medical applications. Thus a statistical model is needed to determine changes in the measurements in a sensible way.

A.2.2 Methods Based on Normal Distribution

In this section, we assume that a sample $X_1, X_2, \ldots, X_n$ is drawn from a normal distribution that has mean μ_x and has variance σ^2, and that another sample $Y_1, Y_2, \ldots, Y_m$ is drawn from another normal distribution that has mean μ_y and the same variance σ^2. The X's can be thought of as being a treatment group, and the Y's as a control group. The effect of the treatment can then be characterized by $\mu_x - \mu_y$. A natural estimate of this value is $\bar{X} - \bar{Y}$, which also proves to be the Maximum Likelihood Estimate (MLE). It is normally distributed:

$$\bar{X} - \bar{Y} \sim N\left[\mu_x - \mu_y, \sigma^2(\frac{1}{n} + \frac{1}{m})\right]$$

If σ^2 were known, a confidence interval for $\mu_x - \mu_y$ could be based on

$$Z = \frac{(\bar{X} - \bar{Y}) - (\mu_x - \mu_y)}{\sigma\sqrt{\frac{1}{n} + \frac{1}{m}}}$$

which follows a normal distribution. The confidence interval would be of the form

$$(\bar{X} - \bar{Y}) \pm z(\frac{\alpha}{2})\sigma\sqrt{\frac{1}{n} + \frac{1}{m}}$$

where z is the z-distribution with $\frac{\alpha}{2}$ DOF. However, since σ^2 is in most cases unknown, it must be estimated from the given data by calculating a *pooled sample variance*,

$$s_p^2 = \frac{(n-1)S_X^2 + (m-1)S_Y^2}{m+n-2}$$

Theorem A: Let samples X and Y be defined as above. Then the statistic

$$t = \frac{(\bar{X} - \bar{Y}) - (\mu_x - \mu_y)}{s_p\sqrt{\frac{1}{n} + \frac{1}{m}}}$$

follows a t distribution with $m + n - 2$ DOF.

Corollary A: Under the assumptions of Theorem A, a $100(1-\alpha)\%$ confidence interval for $\mu_X - \mu_Y$ is

$$(\bar{X} - \bar{Y}) \pm t_{m+n-2}(\alpha/2)s_{\bar{X}-\bar{Y}}$$

We will now introduce a null-hypothesis:

$$H_0 : \mu_X = \mu_Y$$

This means that there is no difference between the distributions of the X's and Y's. If one group is a treatment group and the other a control group, this hypothesis asserts that there is no treatment effect. If there were a treatment effect, the null-hypothesis would be rejected. There are 3 alternatives for the two-sample case:

$$H_1 : \mu_X \neq \mu_Y$$

$$H_2 : \mu_X > \mu_Y$$

$$H_3 : \mu_X < \mu_Y$$

The first of these is a *two-sided alternative*, and the other are *one-sided alternatives*. The quantity that will be used to determine whether or not the null-hypothesis is rejected is

$$\frac{\bar{X} - \bar{Y}}{s_{\bar{X}-\bar{Y}}}$$

The rejection regions for the 3 alternatives which are listed above are

$$\text{For } H_1, |t| > t_{n+m-2}(\alpha/2)$$

$$\text{For } H_2, t > t_{n+m-2}(\alpha)$$

$$\text{For } H_3, t < -t_{n+m-2}(\alpha)$$

It should be noted how the rejection regions are tailored to the particular alternatives, and that knowing the null-distribution of t allows determination of the rejection region for any value of α.

A.2.3 Power

Calculations of power are an important part of planning experiments in order to determine how large sample sizes should be. The power of the two-sample t-test depends on:

1. The real difference, $\Delta - |\mu_X - \mu_Y|$. The larger the difference, the greater the power. Δ denotes the difference between the samples.

2. The significance level α at which the test is done. The larger the level, the more powerful the test.

3. The population standard deviation σ, which is the amplitude of "noise" that hides the "signal". The smaller the sample sizes σ , the greater the power.

4. The sample sizes n and m. The larger the sample sizes, the greater the power.

The necessary sample sizes can be determined from the significance level of the test, the standard deviation and the desired power against an alternative hypothesis,

$$H_1 : \mu_X - \mu_Y = \Delta$$

To calculate the power of a t-test exactly, special tables of the non-central t-distribution are required. However, if sample sizes are sufficiently large, approximate power calculations can be used, which are based on the normal distribution.

Suppose that σ, α and Δ are known and that both sample sizes are n. The test at level α of $H_0 : \mu_X = \mu_Y$ is based on the test statistic

$$Z = \frac{\bar{X} - \bar{Y}}{\sigma\sqrt{2/n}}$$

The rejection region for this test is $|Z| > z(\alpha/2)$, or

$$|\bar{X} - \bar{Y}| > z(\alpha/2)\sigma\sqrt{\frac{2}{n}}$$

The power of the test if $\mu_X - \mu_Y = \delta$ is the probability that the the test statistic falls in the rejection region, or

$$P\left[|\bar{X} - \bar{Y}| > z(\alpha/2)\sigma\sqrt{\frac{2}{n}}\right] =$$

$$P\left[\bar{X} - \bar{Y} > z(\alpha/2)\sigma\sqrt{\frac{2}{n}}\right] + P\left[\bar{X} - \bar{Y} < -z(\alpha/2)\sigma\sqrt{\frac{2}{n}}\right]$$

since both events are mutually exclusive. The RHS probabilities can be calculated by standardizing. For the first one:

$$P\left[\bar{X} - \bar{Y} > z(\alpha/2)\sigma\sqrt{\frac{2}{n}}\right] = P\left[\frac{(\bar{X} - \bar{Y}) - \Delta}{\sigma\sqrt{2/n}} > \frac{z(\alpha/2)\sigma\sqrt{2/n} - \Delta}{\sigma\sqrt{2/n}}\right]$$

$$= 1 - \Phi\left[z(\alpha/2) - \frac{\Delta}{\sigma}\sqrt{\frac{n}{2}}\right]$$

where Φ is the standard normal cdf (Cumulative Density Function). Similarly, the second probability is

$$\Phi\left[-z(\alpha/2) - \frac{\Delta}{\sigma}\sqrt{\frac{n}{2}}\right]$$

Thus, the probability that the test statistic falls in the rejection region is

$$1 - \Phi\left[z(\alpha/2) - \frac{\Delta}{\sigma}\frac{n}{2}\right] + \Phi\left[-z(\alpha/2) - \frac{\Delta}{\sigma}\sqrt{\frac{n}{2}}\right]$$

As Δ moves away from 0, one of these terms will be negligible with respect to the other. For fixed n, this expression can be evaluated as a function of Δ, and vice versa.

A.2.4 Comparing Paired Samples

In many experiments, samples are paired. Since pairing causes the samples to be dependent, the analysis of the previous section does not apply in these cases. First, the design will be studied.

Pairs are denoted by (X_i, Y_i), where $i = 1, \ldots, n$ and it is assumed that the X's and Y's have means μ_X and μ_Y and variances σ_X^2 and σ_Y^2, respectively. Further, it will be assumed that different pairs are independently distributed and that $\mathrm{Cov}(X_i, Y_i) = \sigma_{XY}$. Calculations will be made by using differences $D_i = X_i - Y_i$, which are independent with

$$E(D_i) = \mu_X - \mu_Y$$

$$\begin{aligned}\mathrm{Var}(D_i) &= \sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY} \\ &= \sigma_X^2 + \sigma_Y^2 - 2\rho\sigma_X\sigma_Y\end{aligned}$$

when ρ is the correlation of members of a pair. A natural estimate of $\mu_X - \mu_Y$ is $\bar{D} = \bar{X} - \bar{Y}$, the average difference. From the properties of D_i, it follows that

$$E(\bar{D}) = \mu_X - \mu_Y$$

$$\mathrm{Var}(\bar{D}) = \frac{1}{n}(\sigma_X^2 + \sigma_Y^2 - 2\rho\sigma_X\sigma_Y)$$

On the other hand, an experiment could be done by taking a sample of n X's and an independent sample of n Y's. Then $\mu_X - \mu_Y$ would be estimated by $\bar{X} - \bar{Y}$,and

$$\mathrm{E}(\bar{X} - \bar{Y}) = \mu_X - \mu_Y$$

$$\mathrm{Var}(\bar{X} - \bar{Y}) = \frac{1}{n}(\sigma_X^2 + \sigma_Y^2)$$

If the variances of both estimates are compared, it is clear that the variance of $\bar{D}$ is smaller if the correlation is positive (the X's and Y's are positively correlated).

In those circumstances, pairing is the more effective experimental design. In the simple case in which $\sigma_X = \sigma_Y = \sigma$, the two variances may be more simply expressed as

$$\mathrm{Var}(\bar{D}) = \frac{2\sigma^2(1-\rho)}{n}$$

for the paired case. In the unpaired case, it can be expressed as

$$\mathrm{Var}(\bar{X} - \bar{Y}) = \frac{2\sigma^2}{n}$$

The relative efficiency is

$$\frac{\mathrm{Var}(\bar{D})}{\mathrm{Var}(\bar{X} - \bar{Y})} = 1 - \rho$$

For example, if the correlation is 0.5, a paired design with n pairs of samples yields the same precision as an unpaired design with $2n$ subjects per treatment.

A.2.5 Methods Based on the Normal Distribution

In this section, it is assumed that the differences are samples from a normal distribution with

$$\mathrm{E}(D_i) = \mu_X - \mu_Y = \mu_D$$

$$\mathrm{Var}(D_i) = \sigma_D^2$$

However, since σ_D is generally unknown, inferences will be based on

$$t = \frac{D - \mu_D}{s_{\bar{D}}}$$

which follows a t-distribution with $n-1$ DOF. A $100(1-\alpha)$ confidence interval for μ_D is

$$\bar{D} \pm t_{n-1}(\alpha/2)s_{\bar{D}}$$

A two-sided test of the null-hypothesis $H_0 : \mu_D = 0$ (for testing no treatment at all) at level α has the rejection region

$$|\bar{D}| > t_{n-1}(\alpha/2)s_{\bar{D}}$$

If n (the sample size) is large, the approximate validity of the confidence interval and hypothesis test follows from the central limit theorem. If n is small and the true distribution of the differences is far from normal, the stated probability levels may considerably differ.

A.2.6 Fishing expeditions

A problem that sometimes flaws observational studies, and controlled experiments as well, is that they engage in "fishing expeditions". For example, consider a hypothetical study of the effects of birth control pills. In such a case, it would be impossible to assign women to a treatment- or a placebo group, but a randomized study might be conducted by carefully matching controls to treatments on such factors as age and medical history. The two groups would be followed up on some time, with many variables recorded for each subject such as blood pressure, psychological measures, and incidences of various medical problems. After termination of the study, the two groups might be compared on each of these variables, and it might be found, say, that there is a "significant difference" in the incidence of melanoma. The problem with this "significant finding" is the following.

Suppose that 100 independent two sample t-tests are conducted at the .5 level and that, in fact, all the null hypotheses are true. We would expect that 5 of the tests would produce a "significant" result. However, as a collection they do not simultaneously have $\alpha = .05$. The combined significance level is the probability that at least one of the null hypotheses is rejected:

$$\begin{aligned}\alpha &= \mathrm{P}\{\text{at least one } H_0 \text{ rejected}\} \\ &= 1 - \mathrm{P}\{\text{no } H_0 \text{ rejected}\} \\ &= 1 - .95^{100} \\ &= .994\end{aligned}$$

Thus, with very high probability, at least 1 "significant" result will be found, even if all the null hypotheses are true.

There are no simple cures for this problem. One possibility is to regard the results of a fishing expedition as merely providing suggestions for experiments. Alternatively, the data could be split randomly into halves, one half for fishing in and the other half to be locked safely away, unexamined. "Significant" results from the first half could then be tested on the second half. A third alternative is to conduct each individual hypothesis test at a small significance level. To see how this works, suppose that all null hypotheses are true and that each of n null hypotheses is tested at level α. Let R_i denote the event that the ith null hypothesis is rejected, and let α^* denote the overall probability of a type I error (see [12]). Then

$$\begin{aligned}\alpha^* &= \mathrm{P}\{R_1 \vee R_2 \vee \ldots \vee R_n\} \\ &\leq \mathrm{P}\{R_1\} + \mathrm{P}\{R_2\} + \cdots + \mathrm{P}\{R_n\} \\ &= n\alpha\end{aligned}$$

Thus, if each of the n null hypotheses is tested at level α/n, the overall significance level is less than or equal to α. This is often called the **Bonferroni method**.

A.3 Analysis of variance

A.3.1 Introduction

The previous section was concerned with the analysis arising from experiments with two samples. However, often experiments involve more than two experiments. They may compare several treatments. For example, in one experiment several treatments can be the testing for multiple drugs, and age. This chapter is an introduction to the statistical analysis of such experiments. The discussed methods are called "analysis of variance" (ANOVA). Two designs will be considered: the one-way and the two-way layout. Also, some methods based on the normal distribution will be developed.

A.3.2 The One-Way Layout

A **one-way layout** is an experimental design in which independent measurements are made under each of several treatments. The introduced techniques are thus generalizations of the techniques for comparing 2 independent samples, that were covered in the previous chapter.

A.3.3 Normal Theory: the F Test

Let Y_{ij} = the jth observation of the ith treatment. The model is based on the assumption that observations are corrupted by random errors, and that the error in one observation is independent of the errors in the other observations. The statistical model is

$$Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

Here μ is the overall mean, α_i is the differential effect of the ith treatment, and ϵ_{ij} is the random error in the jth observation under the ith treatment. The errors are assumed to be independent, and are assumed to be normally distributed with mean zero and variance σ^2. The α_i are normalized:

$$\sum_{i=1}^{I} \alpha_{ij} = 0$$

The expected response to the ith treatment is $\mathrm{E}(Y_{ij}) = \mu + \alpha_i$. Thus, if $\alpha_i = 0$, for $i = 1, \ldots, I$, all treatments have the same expected response, and, in general, $\alpha_i - \alpha_j$ is the difference between the expected values under treatments i and j. The analysis of variance is based on the following identity:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(Y_{ij} - \bar{Y}_{..})^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}(Y_{ij} - \bar{Y}_{i.})^2 + J\sum_{i=1}^{I}(\bar{Y}_{i.} - \bar{Y}_{..})^2$$

where

$$\bar{Y}_{i\cdot} = \frac{1}{J}\sum_{j=1}^{J} Y_{ij}$$

is the average of the observations under the ith treatment and

$$\bar{Y}_{\cdot\cdot} = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} Y_{ij}$$

is the overall average. The terms appearing in the first identity above are called "sums of squares", and the identity may symbolically be expressed as

$$\mathrm{SS_{TOT}} = \mathrm{SS_W} + \mathrm{SS_B}$$

In normal language, this means that the *total* sum of squares equals the sum of squares *within* groups plus the sum of squares *between* groups. This reflects that $\mathrm{SS_W}$ is a measure of the variation of the data within the treatment groups and that $\mathrm{SS_B}$ is a measure of the variation of the treatment among or between treatments.

Lemma A: Let X_i, where $i = 1, \ldots, n$, be independent random variables with $\mathrm{E}(X_i) = \mu_i$ and $\mathrm{Var}(X_i) = \sigma^2$. Then

$$\mathrm{E}(X_i - \bar{X})^2 = (\mu_i - \bar{\mu})^2 + \frac{n-1}{n}\sigma^2$$

where

$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mu_i$$

Theorem A: Under the assumptions for the model stated at the beginning of this section,

$$\begin{aligned}
\mathrm{E}(\mathrm{SS_W}) &= \sum_{i=1}^{I}\sum_{j=1}^{J} \mathrm{E}(Y_{ij} - \bar{Y}_{i\cdot})^2 \\
&= \sum_{i=1}^{I}\sum_{j=1}^{J} \frac{J-1}{J}\sigma^2 \\
&= I(J-1)\sigma^2
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}(\mathrm{SS_B}) &= J\sum_{i=1}^{I} \mathrm{E}(\bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot})^2 \\
&= J\sum_{i=1}^{I} \left[\alpha_i^2 + \frac{(I-1)\sigma^2}{IJ}\right]
\end{aligned}$$

$$= J \sum_{i=1}^{I} \alpha_i^2 + (I-1)\sigma^2$$

Since $\mathrm{SS_W}$ may be used to estimate σ^2, the estimate is

$$s_p^2 = \frac{\mathrm{SS_W}}{I(J-1)}$$

which is unbiased. The subscript p indicates that it is *pooled*. Estimates of σ^2 from the I treatments are pooled together, since $\mathrm{SS_W}$ can be written as

$$\mathrm{SS_W} = \sum_{i=1}^{I} (J-1)s_i^2$$

where s_i^2 is the sample variance in the ith group.
If all the α_i equal 0, then the expectation of $\mathrm{SS_B}/(I-1)$ is also σ^2. In this case, that means that $\mathrm{SS_W}/[I(J-1)]$ and $\mathrm{SS_B}/(I-1)$ should be about equal. If some of the α_i are nonzero, $\mathrm{SS_B}$ will be inflated.

Theorem B: If the errors are independent and follow a $\sim \mathrm{N}(0, \sigma^2)$ distribution, then $\mathrm{SS_W}/\sigma^2$ follows a $\chi^2_{I(J-1)}$ distribution, and is independent of $\mathrm{SS_W}$.

A widely used form in which sums of squares are used, is a representation in the form of an *analysis of variance table*. An example of such a representation is the following table:

Source	*df*	*SS*	*MS*	*F*
Labs	6	.125	.021	5.66
Error	63	.231	.0037	
Total	69	.356		

In this table, $\mathrm{SS_W}$ is the sum of squares due to error, and $\mathrm{SS_B}$ is the sum of squares due to lab differences. *MS* stands for *Mean Square*. It equals the sum of squares divided by the DOF. The column named F displays the F-statistic for testing the null-hypothesis (there is no difference among the labs). The F-statistic has 6 and 60 df and has a value of 5.66.

Appendix B

How to use the Wavelet tool-box for Matlab

The Wavelet Toolbox for use with Matlab is a set of Matlab functions and a graphical environment which can be used to compute, visualize and program wavelets in the Matlab computer environment. Supported in this tool-box are 1D and 2D discrete wavelets, 1D and 2D wavelet packets, continuous 1D wavelets.

Graphical environment

The graphical user environment can be started by typing `wavemenu` at the Matlab prompt. When this is done a window will appear at the screen, with 8 buttons which are labeled 'Wavelet 1-D', 'Wavelet 2-D', 'Wavelet Display', 'Wavelet Packet 1-D', 'Wavelet Packet 2-D', 'Wavelet Packet Display', 'Continuous Wavelet 1-D' and 'Close'. The only part of the graphical environment which has been used in this project is the 'Wavelet 2-D' part, and therefore only this part of the graphical environment will be explained in more detail.
When the 'Wavelet 2-D' button is clicked, a new window will appear.

This window is divided into 4 areas: a menu bar, an image display part, an image part selection tool bar (at the bottom of the window) and an operations panel at the right part of the window. The image display part is also divided into 4 parts. Its upper left region displays the original image, the bottom left region is the reconstructed image. The bottom right part contains the image in the wavelet domain, and the upper right part of the image display part optionally shows a user-selected part of the wavelet representation of the image. This user-selected part is either the coarsest scale approximation, or a set of detail coefficients (either horizontal, vertical or diagonal channel) at a chosen level. However, by default this section of the image display part is empty, and the user first has to click in a bottom right part of the decomposition followed by clicking the 'Visualize' button. Other options instead of clicking the 'Visualize' button are the 'Full Size'

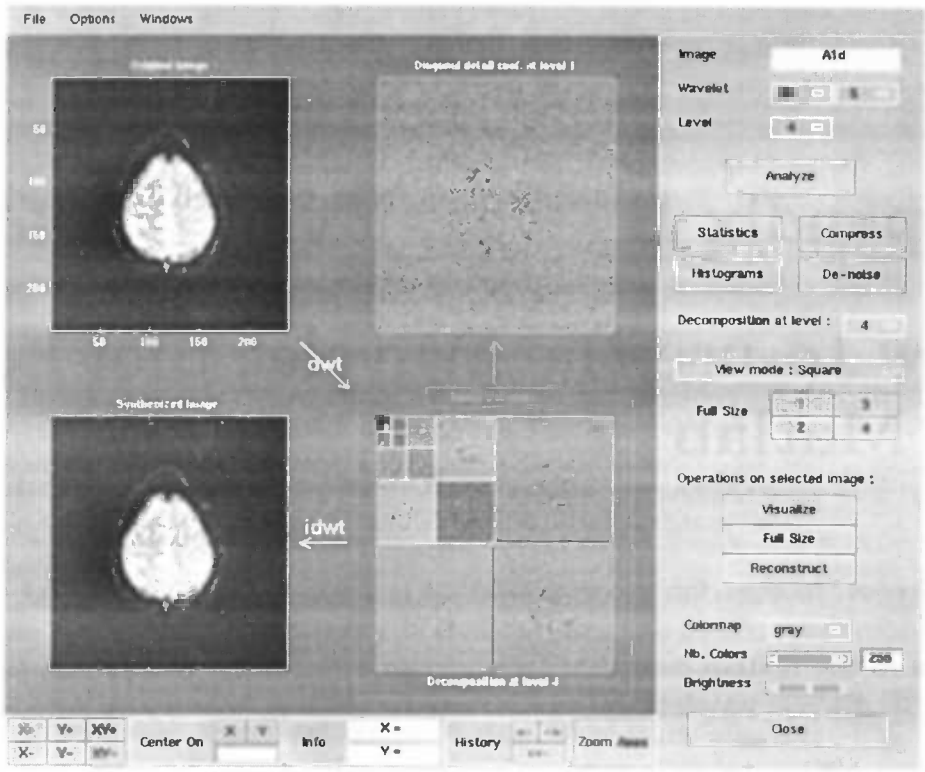

Figure B.1: The Wavelet 2-D user work area

and 'Reconstruct' button. The first one of these two choices results in displaying the selected region of the decomposition to be drawn in the entire image display region (and making it the only visible view on the data), and the second choice reconstructs an image using only the selected subset of the decomposition by temporarily discarding all the other detail coefficients and therefore enables the user to obtain insight in the behavior of the selected detail component when an operation is applied which modifies that component.
Furthermore, the user can adjust the way the data is displayed by adjusting the overall brightness of the visualizations as well as the number of distinct colors used in the plots with a slider. Also different color-maps (predefined by the tool-box) can be used to gain insight in subtle image features which might not have been noticed by the user when the same color-map would have been used throughout the data analysis process.
Other features of the Wavelet 2-D user work area are the 'Statistics', 'Compress', 'Histograms' and 'De-noise' button. All these buttons cause the creation of a new window with a layout depending on which button of the four has been pressed. A short summary of the purpose of these four buttons:

1. Statistics A window pops up containing statistical information about the

selected image (or image-part). For example, the average and minimal values are displayed.

2. Histograms A window pops up with histograms of the selected image.

3. Compress A window pops up with the purpose to compress the wavelet signals by discarding non-relevant detail coefficients (a percentile can be selected) while retaining maximum energy in the channels.

4. De-noise A window pops up in which the user can alter some parameters to answer the question which detail coefficients are to be classified as carrying noise only, and which coefficients carry signal information.

All windows have the property that the result can be viewed immediately after modifying any of the parameters, and discarding the resulting image if the modification was unsatisfying. However if the modification was successful, the user can choose to make the modification permanent and therefore update the according image in the Wavelet 2-D user work area.

Command-line driven functions

The Wavelet tool-box provides the user with a lot of functions, and the ones that were used most in this report are listed below, with a short explanation.

1. Analysis-decomposition functions:
   - `dwt2` Single-level discrete 2-D wavelet transform
   - `wavedec2` Multi-level 2-D wavelet decomposition
2. Synthesis-reconstruction functions:
   - `idwt2` Single-level inverse discrete 2-D wavelet transform
   - `waverec2` Multi-level 2-D wavelet reconstruction
   - `wrcoef2` Reconstruct single branch from 2-D wavelet coefficients
   - `upcoef2` Direct reconstruction from 2-D wavelet coefficients
3. Decomposition structure utilities:
   - `detcoef2` Extract 2-D detail coefficients
   - `appcoef2` Extract 2-D approximation coefficients
   - `upwlev2` Single-level reconstruction of 2-D wavelet decomposition

Further information (including the required and optional parameters for each function) can be obtained by entering `help` *<function>* at the Matlab prompt, with *<function>* replaced by any of the function names which are listed above.

Bibliography

[1] Wadsworth & Brooks/Cole, editor. *Mathematical statistics and data analysis.* Pacific Grove, CA., 1988.

[2] Robert W. Cox. *MCW AFNI — User Manual.* Medical College of Wisconsin, December 1996.

[3] G. Kerkyacharian D.L. Donoho, I.M. Johnstone and Picard. Wavelet shrinkage: Asymptotia. In *J. Roy Statistical Soc., Series B*, volume 57, pages 269–301, 1995.

[4] A. P. Holmes. *Statistical issues in functional brain mapping.* PhD thesis, Department of Statistics, University of Glasgow, December 1994.

[5] J.V. Hajnal, R. Myers, A. Oatridge, J.E. Schwieso, I.R. Young and G.M. Byder. Artifacts due to stimulus correlated motion in functional imaging of the brain. *Magn. Reson. Med.*, 31, 1994.

[6] K.J. Friston, J. Ashburner, C.D. Frith, J.-B. Poline, J.D. Heather, R.S.J. Frackowiak. Spatial registration and normalization of images. *Human brain mapping*, 2:165–189, 1995.

[7] P. Neelin K.J. Worsley, S. Marrett and A.C. Evans. Searching scale space for activation in PET images. *Human Brain Mapping*, 4(1):74–90, 1996.

[8] C.T.W. Moonen. Imaging of human brain activation with functional MRI. *Biological Psych.*, 37, 1995.

[9] P. Nijland. *Analysing brain activity using Statistical Parametric Mapping.* PhD thesis, R*u*G, 1999.

[10] M.S. Hinks P.A. Bandetti, E.C. Wong and R.S. Tifofsky. Time course EPI of human brain function during task activation. *Magn. Reson. Med.*, 25:390–397, 1992.

[11] J.B. Poline and B.M. Mazoyer. Analysis of individual brain activation maps using hierarchical description and multiscale detection. In *IEEE Trans. Med. Imag.*, volume 13, pages 702–710, Aug. 1994.

[12] J.A. Rice. *Mathematical statistics and Data analysis*, chapter 12. Duxbury Press, 2nd edition, 1995.

[13] J.B.T.M. Roerdink. Computerized tomography and its applications: a guided tour. *Nieuw archief voor Wiskunde*, 10(3):277–308, November 1992.

[14] A.S. Nayak S. Ogawa, T.M. Lee and P. Glynn. Oxygenation sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magn. Reson. Med.*, 14:67–78, 1990.

[15] R.L. Savoy. A two day workshop on functional magnetic imaging, June 16-17 1996. Boston MA.

[16] Michael Unser Urs. E. Ruttiman and Robert R. Rawlings. Statistical analysis of functional MRI data in the wavelet domain. In *IEEE Trans. Med. Imag.*, volume 17, pages 142–154, April. 1998.

[17] B. Douglas Ward and Robert W. Cox. *MCW AFNI 2.20 — Auxiliary programs.* Medical College of Wisconsin, January 1998.