Bayesian Signal Processing and Interpretation of Brain Scans

M.Sc.E.E. Ulrik Kjems

February 28, 1998

Bayesian Signal Processing and Interpretation of Brain Scans

The present report is the ph.d. dissertation describing work accomplished at the Section for Digital Signal Processing, department of Mathematical Modelling at the Technical University of Denmark in the period February 1995–February 1998. The report contains three parts: The first part describes the fundamentals of brain imaging using PET and MRI scanners; the second describes Bayesian image restoration procedures and lays out the mathematical foundation for the use of object prior models for performing "Bayesian" image noise suppression. Methods for parameter estimation in non-normalized (Gibbs) distributions in random field image models are considered. A generalized learning algorithm, the so called Boltzmann Learning rule, is investigated and a modification of the learning rule for proper treatment of scale parameters is suggested, allowing the training of all-pole models that are guaranteed stable. The third part of the thesis describes a novel method that has been developed for non-linear standardization (warping) of MRI brain image volumes. The method is tested in a multi-subject $[^{15}O]$water PET functional activation study of saccadic eye movements, and demonstrates superior performance when compared to a conventional 12-paramter affine registration method: Less voxels are masked out in the final activation analysis due to registration errors, and the estimated activation signal to noise ratio is increased by approximately 30 %.

Bayesiansk Signalbehandling og Fortolkning af Hjerneskan

Nærværende rapport er Ph.D. afhandlingen over arbejde udført ved Sektionen for Digital Signal Behandling, Instititut for Matematisk Modellering ved Danmarks Tekniske Universitet i perioden februar 1995 til og med februar 1998. Denne rapport indeholder tre dele: Den første giver en grundlæggende beskrivelse af billeddannende teknikker af hjernen ved brug af PET og MRI skannere; anden del beskriver Baeysianske billed-restaurerings metoder og beskriver den matematiske teori for at udføre den "Baysianske" støjreduktion i hjerne-billeder. Metoder til parameter-estimation i ikke-normaliserede (Gibbs) fordelinger beskrives. En generaliseret indlæringsregel—den såkaldte Boltzmann-indlæringsregel—undersøges og der foreslås en modifikation af indlæringsregelen som giver en mere korrekt behandling af skala-parametre, og som giver mulighed for indlæring af rekursive modeller (eng. *all-pole models*) som er garanteret stabile. Tredje del beskriver en ny metode som er blevet udviklet til at udføre ikke-lineære stardardisering (eng. *warp*) af MRI billed-volumener. Metoden er testet på et flerpersoners $[^{15}O]$-vand PET funktionelt aktiveringsstudie af sakkadiske øjenbevægelser, og giver forbedrede analyseresultater sammenlignet med en konventionel 12 parameter affin registreringsmetode: Færre voxels bliver masket bort i den endelige aktiveringsanalyse, og det resulterende estimerede aktiverings signal støj-forhold bliver forbedret med ca. 30 %.

Contents

Preface

This dissertation is the description of work accomplished at the Section for Digital Signal Processing, department of Mathematical Modelling at the Technical University of Denmark in the period February 1995–February 1998. My advisors during the project period have been associate professor Lars Kai Hansen (primary advisor) and associate professor Jan Larsen (co-advisor), both at the Section for Digital Signal Processing at the Department of Mathematical Modelling. The subject of the thesis is "Bayesian Signal Processing", i.e. methods for incorporating prior knowledge of an object into signal processing applications with the intend to reduce noise factors in studies of the functional human brain.

The intention of this thesis is on one hand to provide an introduction into the field of image processing and noise reduction in brain imaging and on the other hand to describe in detail some of the novel image processing algorithms that have been developed in the last three years' work.

Image distortion and degradation are phenomena frequently encountered in image acquisition systems. The aim of the present thesis is to provide a discussion of a variety of techniques for image noise reduction with a particular interest in the application to MRI and PET images of the living human brain.

Several obstacles are encountered in the reconstruction process. One is the computational size of the problem—a high resolution MR scan of a human head will typically contain 16 million voxels. This enforces a hard limit on the computational complexity of algorithms for them to be usable in practice.

The discipline of functional brain imaging contains hosts of image processing subproblems, each of which can be refined and perfected separately; this makes the task of identifying the weak link in the chain of processing steps a difficult one indeed. The present report by no means attempts to cover all aspects involved. Rather, the more general theory and broader considerations will be applied and focused on whenever possible. The MRI segmentation algorithm utilized in some of the examples is by no means optimal, which (hopefully) should be of minor importance for the conceptual clarity of the modeling techniques presented.

In this thesis Bayesian Image restoration is thought mainly as the utilization of prior knowledge in the image reconstruction process. The human visual perception system displays excellent performance in this discipline, being able to utilize previously attained experience about an object recognized in an environment degraded by factors like noise, illumination, blurring, perspective etc. The objective in the following will be to investigate the mathematical representations of such prior information about the object being imaged. The motivation is not to imitate the principles of human image perception, but to develop the mathematical and statistical foundation for the application of prior models to the restoration and noise reduction problem.

In the first Chapter, we will take a look at the factors involved in creating the PET and MRI brain scans. Knowledge about the imaging systems as well as the system being imaged (i.e. the brain) is crucial for creating image models for a specific image modality. Chapter two is devoted to the statistical modeling of images, with the specific approach of regarding them as random fields. The major part of my work in this field has been concentrated on the parameter estimation problem in these models, which is the subject of main emphasis in this Chapter. The third Chapter describes a novel method that has been developed for performing non-linear registration, aka. 'warping', of human brains. The objective is to improve comparison of functional scans across individuals. An application of the algorithm to a saccadic eye movement experiment demonstrated that the signal to noise ratio was improved by as much as 40 % when compared to a classical affine normalization method. Finally, the programs for non-linear registration have been documented and made publicly available. The documentation (in the form of Unix `man`-pages) is included in the Appendix.

Many people have been helpful during the three year course of my project. I would like to thank my advisors Lars Kai Hansen and Jan Larsen for valuable coaching, supervision and intelligent input. I would also like to thank all the co-workers and collaborators who have treated me with great hospitality during my visits to their laboratories: dr. Chin-Tu Chen and dr. Yali Amit at the University of Chicago have provided vital and discussion input during the development of the non-linear registration algorithm. The team at the VAPET Center, University of Minnesota, Minneapolis has been extremely kind during my stay there, especially dr. Stephen Strother, whom I owe a great deal the good ideas for functional evaluation of the warping algorithm. The people at the SALK institute, dr. Terrence Sejnowski and dr. Michael McKeown have also offered great hospitality letting me enjoy the great atmosphere at the CNL laboratory and dr. Terry Jernigan at the Brain Image Analysis Laboratory for enabling my visit in San Diego. Furthermore I wish to thank dr. Alan Evans for his benign hospitality; my visit to the BIC center provided me with much of the inspiration for the mutual information based cost functions; I was very impressed with the McBic group's professionalism and proficiency at handling many of the image processing problems in connection with MRI imaging. Last but not least, my colleague students (Niels Mørch, Mads Hintz-Madsen and especially Peter Philipsen in collaboration with whom much of the work with parameter estimation has been performed) have been extremely supportive through the entire period.

This work was supported in part by the Technical University of Denmark, the Human Brain Project grant P20 MH57180, the Danish Research councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center (CONNECT), the SLOAN Center for Theoretical Neurobiology at the SALK institute CA, and Knud Højgaards fond, Denmark.

Lyngby, February 1998

Chapter 1

Brain Imaging

Scanners able to capture in vivo images of the human brain include the Positron Emission Tomography (PET) scanner, X-ray Computed Tomography (CT), (Single Positron Emission Tomography (SPECT), Magneto-encelography (MEG), Electro-Encelography (EEG), nuclear Magnetic Resonance Imaging (MRI) with functional MRI (fMRI) as the most recent development. Each of these scanning techniques produces images of the spatial distribution of selected conditions in the brain. Some modalities depict tissue dependent information (MRI, CT), others reveal the spatial distribution of an induced tracer (SPECT, PET, MRI), electrical current and induced magnetic field (EEG and MEG), blood metabolism (PET and SPECT) and blood oxygenation (BOLD with fMRI).

Some of these factors depend on the local activity in the brain. It is thus possible to measure and quantify changes in brain activation induced by an external control paradigm. For this reason these functional scans have attained considerable interest for investigating the workings of the human brain.

This report will mainly focus on the use of the PET scanning technique as the functional image modality and MRI scans as the source of structural information although the theory presented is also applicable to other sources of images. Figure 1.1 shows 3 orthogonal slices of a PET and MRI volume of the same subject, co-registered for comparison.

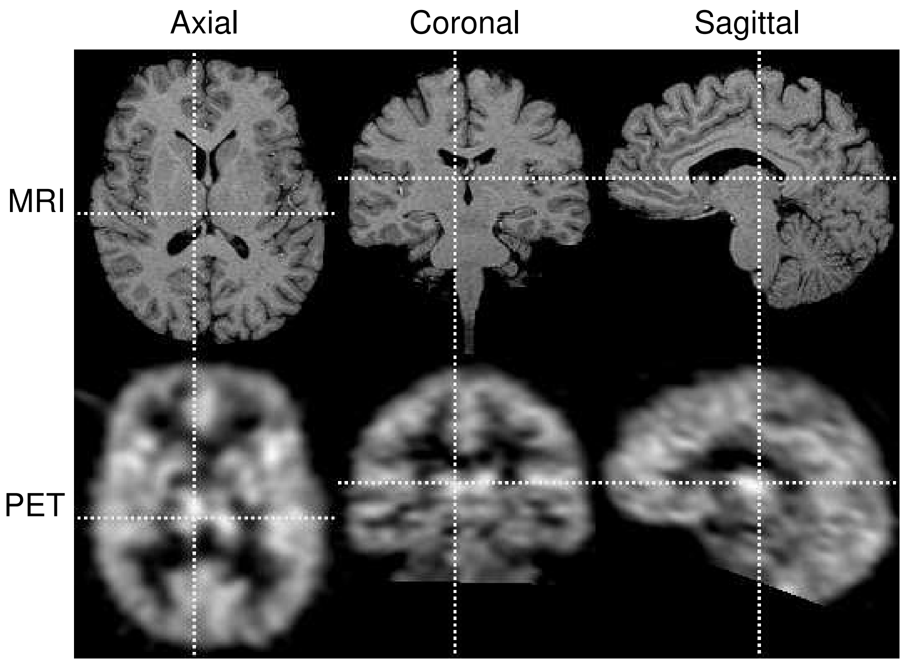


Figure 1.1: Example of co-registered MRI and PET image volumes of the same subject. The MRI volume has been stripped so that only brain tissue is displayed.

1.1 The PET scan

A PET scanner is basically a set of scintillation detectors placed in rings around the subject's head (or another object of interest) as illustrated in Figure 1.2. A bolus of radioactive tracer is either injected intravenously or inhaled by the subject. The tracer will enter into the blood supply of the brain and, depending on the regional blood flow and tracer accumulation, distribute itself into the brain volume.

The scanner relies on the fact that certain isotopes decay by emitting a positron which quickly annihilates with a nearby electron, emitting two gamma quanta with almost opposite flight-paths (the flight paths are not exactly opposite because the momentum of the emitted positron is conserved). The path of the positron from emission until recombination results in an inherent blurring of the image with a blurring kernel identical to the probability distribution of the distance traveled. This distribution depends on the tracer used and on local tissue characteristics, and is in the order of 5 mm for $\left[^{15}\text{O}\right]$water.

Common isotopes used as tracers are $\left[^{15}\text{O}\right]$ bound to water (producing blood-flow related images) and $\left[^{18}\text{F}\right]$ bound to de-oxyglucose (producing images of glucose metabolism). It is possible to label many different compounds making it possible to map out oxygen consumption, locations of receptors for neuro-transmitters and hormones.

The pair of emitted gamma rays is detected in the detector rings surrounding the subject. Using accurate time-stamping, the pairwise simultaneous detections are counted and collected into the PET sinogram. The sinogram is a collection of projections of the radio-activity inside the scanner. Tissue absorption and scatter of gamma rays distort the sinogram and have to be corrected for prior to reconstructing the actual image from the sinogram projections. These corrections to the sinogram are performed by acquiring a CT scan prior to the bolus injection: A source of gamma radiation is rotated around the subject's head so that the attenuation of the gamma source in the subject's head as a function of source angle can be measured, and used to correct the sinogram. A computer program finally transforms the sinogram into the actual three dimensional image volume. Since the sinogram image mathematically can be regarded as a Radon transformed image, the image reconstruction process is referred to as the inverse Radon transformation.

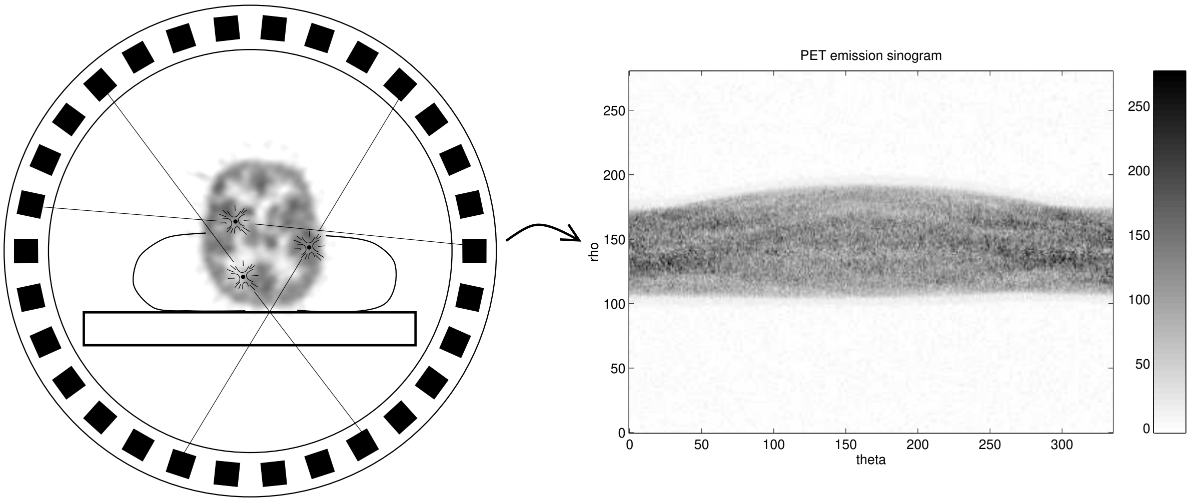


Figure 1.2: The PET scanner consists of parallel rings of scintillation detectors counting pairs of emitted gamma rays.

1.1.1 The Radon Transform

The PET scanner contains multiple parallel rings of detectors (the PET scanner at the National University Hospital, Copenhagen contains 16 detector rings) so the full sinogram is a multidimensional data structure, unlike the single plane sinogram depicted in Figure 1.2. The full sinogram contains projection information of lines in the planes of the detector rings as well as of lines connecting detectors from different rings.

The sinogram is related to the image by the Radon transform, which expresses that the sinogram (the total gamma count of the corresponding detector pair) is proportional to the line integral through the object. For simplicity the following equations are expressed for the two-dimensional case. The line integral

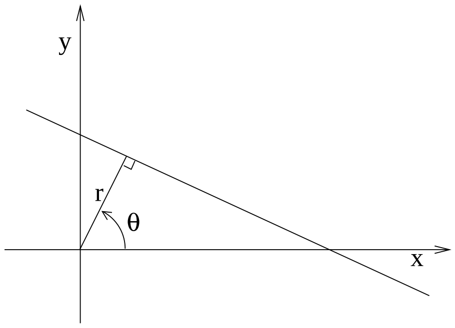


Figure 1.3: r and θ define a line in the image plane.

is parameterized by the transform parameters (r, θ)

$$g(r,\theta) = \iint_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta + r)\,dx\,dy. \tag{1.1}$$

The definition of the parameters is shown in Figure 1.3. It is possible to perform an analytical inversion of the Radon transformation based on the Central Projection Theorem, see e.g. [1, Ch. 10.5]. A number of derived algorithms for analytical reconstruction exist of which the Filtered Back-projection method is a well-known example.

The extension of the Radon theory to three dimensional signals requires working with plane integrals as the observed projections. This means that the observed line integrals need to be interpolated and re-binned to estimate the appropriate plane integrals. The scanner geometry does not approximate the assumptions of the three dimensional Radon transform very well because only a subset of the possible plane integrals are adequately sampled. With the evolution of computational capability, computationally more demanding algorithms have attained popularity, regarding the reconstruction problem instead as a problem of solving a large set of linear equations. These method are known as the algebraic reconstruction techniques.

Being a linear operation, the observed discretely sampled projections (i.e. the Radon transform) may be expressed as a matrix multiplication of the observed object with a projection matrix $\mathbf{A}$

$$\boldsymbol{g} = \mathbf{A}\boldsymbol{f} \tag{1.2}$$

The projections are encoded in a single vector with $g_i = g(r_i, \theta_i)$ so that all possible detector pair combinations are represented. The object is encoded in the vector[1] $\boldsymbol{f}$ using either a set of basis functions where the elements of $\boldsymbol{f} = \{f_j\}$ make up the coefficients

$$f(x,y,z) = \sum_i f_j \phi_j(x,y,z) \tag{1.3}$$

or, more commonly, each f_j represents an image voxel so that the j'th column of the projection matrix $\mathbf{A}$ is the projection of the single voxel f_j onto all possible detector pairs.

Different iterative techniques for solving (1.2) are available: EM (Expectation Maximization), conjugate gradient, ART (Algebraic Reconstruction Technique) see e.g. [2, 1]. These methods are categorized into the Minimum Norm Least Squares solutions (MNLS) because they approximate the solution using

$$\boldsymbol{f}^{+} = \mathbf{A}^{+}\boldsymbol{g} \tag{1.4}$$

where $\mathbf{A}^{+}$ is the generalized inverse of $\mathbf{A}$ since this matrix cannot be expected to have full rank. The solution obtained is called Minimum Norm Least Squares solution (MNLS) because the solution $\boldsymbol{f}^{+}$ is the solution of (1.2) that has the smallest norm $||\boldsymbol{f}^{+}||_2$.

The main practical limitation of the algebraic approach is the size of the projection matrix $\mathbf{A}$: to construct a 128×128 size image from 16000 projections, the projection matrix size is 16000×16384. In practice $\mathbf{A}$ may need to be computed on-line to save memory; in any case the sparse structure of the matrix should be exploited using sparse matrix techniques [2].

[1] A three-dimensional image volume $d(i,j,k), i = 0 \ldots I-1, j = 0 \ldots J-1, k = 0 \ldots K-1$ may be represented as a vector with a linear mapping of voxels (: volume image elements) $d(i,j,k) = \boldsymbol{d}_{1+i+Ij+IJk}$

The advantages of the matrix formulation of the reconstruction problem is that the geometry of the scanner can be modeled better, and we can introduce a noise term into the equation so that noise characteristics are modeled separately:

$$\boldsymbol{g} = \mathbf{A}\boldsymbol{f} + \boldsymbol{n}, \qquad \boldsymbol{n} \in \mathcal{N}(0, \Sigma_n) \tag{1.5}$$

By further adding a prior distribution to the image $\boldsymbol{f} \in \mathcal{N}(0, \Sigma_f)$ this brings the problem into Wiener filter (Bayesian) type of problems which we shall investigate later in Chapter 2.

1.1.2 Noise Modeling in PET

In this section we consider a statistical model of the noise in the reconstructed PET image. Such models are derived from the fact that we are counting spontaneous events of a nuclear process. Spontaneous nuclear events are modeled quite precisely by the Poisson process: A Poisson process arises when one has N potential events, $N \to \infty$, each with identical constant, independent probability p, $p \propto \lambda/N \to 0$ for an event happening within a fixed length of time. For example, in radioactive matter we have a practically infinite number of nuclei and the spontaneous decays of individual nuclei are independent events that have identical and constant probabilities. So a Poisson process is a generator of events, and the Poisson distribution arises when counting events within a period of time. The Poisson distribution $\mathcal{P}(\lambda)$ has a single parameter λ which has the unit of events/measure, measure being length, time, area, volume.

The Poisson process has the following property: Let $p(x,t)$ be a function describing each event at location x_i, time t_i with a delta function $p(x,t) = \sum_i \delta(x - x_i, t - t_i)$, now

$$\int_{(t,x)\in(T,X)} p(x,t)\, dx\, dt \in \mathcal{P}(\lambda), \text{ where } \lambda = \int_{(t,x)\in(T,X)} \lambda(x)\, dx\, dt \tag{1.6}$$

In the above equation, a radioactive object is described by the distribution of Poisson intensities $\lambda(x)$. For a PET scan, the integral over time represents the (e.g. 90 second) time window used for counting gamma pairs after bolus injection. Each pair of detectors performs one integral of (1.6)

$$\begin{aligned} g(r,\theta) &= \int_{s=-r}^{r} \lambda(x,y) p_{\mathrm{d}}(x,y)\, ds \in P(\lambda(r,\theta)) \\ x &= r\cos\theta + s\sin\theta, \quad y = r\sin\theta - s\cos\theta \end{aligned} \tag{1.7}$$

where $p_{\mathrm{d}}(x,y)$ is the probability of detection of a gamma pair emitted from location (x,y). The number of gamma photons hitting a detector pair should approximate the Poisson distribution. For a Poisson distributed variable we have $E(x) = \lambda$ and $V(x) = \lambda^2$, and we know that a Poisson variable with a high expectation is well approximated by the Gaussian distribution with matching mean and variance. The Poisson noise model is weakened by the fact that the scatter correction relies on a (noisy) CT measurement with the end result that the sinogram elements statistics differ from the Poisson distribution. Detector dead-time of the scanner also effects the distribution. In the absence of better models, however, it is should still be fair to assume that each sinogram element $g = g(r,\theta)$ is Poisson distributed. Applying the Gaussian approximation to the observed sinogram we get $\boldsymbol{g}_{\text{obs}} = \boldsymbol{g}_{\text{true}} + \varepsilon$ where $\varepsilon \in \mathcal{N}(0, \Sigma_\varepsilon)$ and $\Sigma_\varepsilon = \text{diag}(g^2)$.

We may transform this model into a covariance matrix for the reconstructed image. For the MNLS solution $\boldsymbol{f}^+ = \mathbf{A}^+\boldsymbol{g}$ we have

$$\Sigma_{f+} = \mathbf{A}^+ \Sigma_\varepsilon \mathbf{A}^{+T}. \tag{1.8}$$

This indicates that, given a reconstructed PET scan and knowledge of the scanner geometry but with no sinogram present, we may build approximations to the noise covariance through estimation of $\mathbf{A}^+$. A major limitation to this approach is, once again, the size of the matrices $\mathbf{A}^+$ and Σ_{f+}.

1.1.3 PET Noise Structure

Some heuristic approximations can be made about the nature of the covariance relationship between voxels in the reconstructed image. First of all we may assume that the reconstruction noise has a lower amplitude in the central part of the scanner's field of view simply because more lines of detector pairs cross there, so more information is present in the sinogram.

The analytical inverse Radon transformation method 'Filtered Back-projection' consists, as the name suggests, of two steps: A back-projection step, which is basically the operation

$$f_{bp}(x,y) = \int g(r,\theta)\delta(x\cos\theta + y\sin\theta + r)\,dr\,d\theta \tag{1.9}$$

or, in the algebraic formulation

$$\boldsymbol{f}_{bp} = \mathbf{A}^T\boldsymbol{g} \tag{1.10}$$

and a filtering operation which 'un-filters' the error of replacing the inverse Radon transformation with the back-projection. It can be shown that this error is identical to filtering with the isotropic kernel $1/r$. This un-filtering (the so called reconstruction filter) is most conveniently performed applied the Radon domain (see e.g. [1, Ch. 10]). Expressing the un-filtering operation in the discrete case with the matrix $\mathbf{D}$ we have $\boldsymbol{f} = \mathbf{A}^T\mathbf{D}\boldsymbol{g}$ so that $\Sigma_{f+} \simeq \mathbf{A}^T\mathbf{D}\Sigma_\varepsilon\mathbf{D}^T\mathbf{A}^T$. Assuming that $\mathbf{Q} = \mathbf{D}\Sigma_\varepsilon\mathbf{D}^T$ is diagonally dominant (which to some extent will be the case when using reconstruction filters like RAMLAK, Shepp-Logan and the like) we obtain $\Sigma_{f+} \simeq \mathbf{A}^T\mathbf{Q}_{diag}\mathbf{A}$ and we see that the covariance element (i,j) is related to the inner product of the projections (i.e. columns of $\mathbf{A}$) of voxels represented by indices i and j respectively and roughly proportional to the sum of the activities (albeit filtered) on lines passing through both voxels. This in turn suggests that we should expect less correlation as distance between the voxels i and j increases since the inner product of the voxel projections will decrease. Furthermore, we should expect higher correlations in directions where the sum of common projections is high. This means the noise correlation function will be highly dependent on image location. The reconstruction noise will be seen to 'radiate' from the brain as can be seen in Figure 1.4 where the palette has been adjusted to focus on the noise patterns outside the brain.

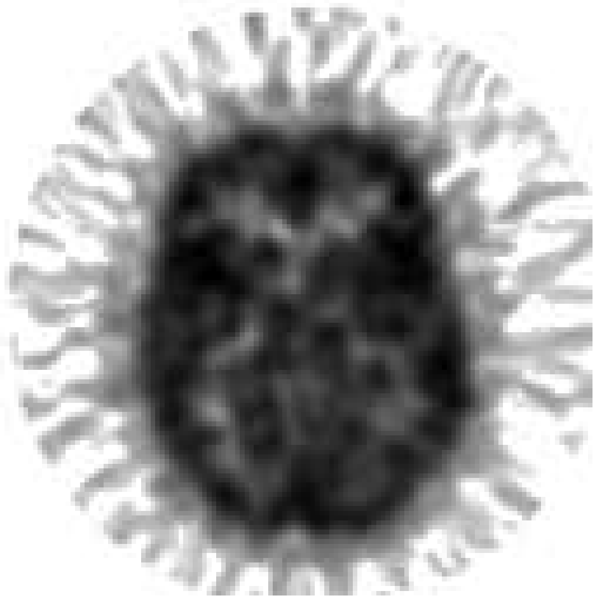


Figure 1.4: A PET scan with visible noise correlation 'radiating' in directions with high counts. The color map has been adjusted to enhance noise appearance.

1.2 The MRI scan

Nuclear Magnetic Resonance Imaging is a non-invasive imaging technique (meaning no tracer is induced and the scanning process is believed to be harmless to the subject), producing anatomical images of high spatial resolution and offers a good signal to noise ratio. The technique is based on the magnetic spin property of hydrogen, each nucleus (proton) can be considered as a small magnetic dipole, which will aligns itself accordingly if a magnetic field is applied. When excited by an externally induced magnetic RF impulse matching the resonant frequency, the dipoles will, in the presence of a magnetic field, perform a precessing of the dipole axis about the direction of the magnetic field lines. The oscillation frequency (the resonant frequency) is directly proportional to the external field strength; for a 1.5 T external field, the frequency falls within the RF band and can be picked up by a coil inside the scanner. The image acquisition proceeds by exciting the hydrogen spins using the RF coil, and subsequently measuring the echo emitted by the spins as they relax into steady state. Spatial localization of the signal is possible through phase and frequency modulation techniques. It turns out that the image sampling procedure can be performed directly in k-space, i.e. Fourier transform space. This means that the Nyquist sampling is obeyed; the raw MRI images are free from aliasing noise. This also means that the sinc interpolator is 'native' for resampling of MRI images.

The MR scanner produces images of local magnitic tissue properties; properties which differ for tissues like white matter (mainly of myelin coated axons, see [3, Ch. 1]), gray matter (where the neuronal nuclei concentrate) and Cerebro Spinal Fluid (which contains a large fraction of water). Depending on the strategy used for RF excitation and read out, known as the *acquisition sequence*, it is possible to control which magnetic properties to weigh in the images. Proton density (pd) images enhance CSF contrast, other modalities measure different time constants related to the relaxation of the hydrogen spins (t1 and t2 weighted images). Good gray/white matter separation is obtained in t1 weighted images. Recent techniques have evolved allowing functional imaging with MR scanners, called fMRI. When neuronal activity increases in a cerebral region, there will be an excess of blood supply yielding an increase in oxy-haemoglobin ('luckily' oxy- and de-oxy-haemoglobin has different magnetic properties) which is measured by the BOLD (Blood Oxygenation Dependant) imaging techniques. Functional MRI has a superior resolution, both spatial and temporal to PET, image sampling rates in the order of 1-2 images per second (single slice recordings) is possible. A problem with fMRI is the the activity signal is harder to interpret compared to PET.

MRI Noise Sources

Figure 1.1 gives an impression of the MRI signal to noise ratio and resolution. There are numerous sources of noise in MR images. First of all, the image acquisition sequence is typically a tradeoff between image resolution and noise level. For t1 weighed images (like the one in Figure 1.1), $1 \times 1 \times 1$ mm is a typical compromize. The spatial correlation structure of the static background noise is well behaved; there are no long range correlations similar to the long range correlations found in PET. Other and more serious noise effects are labelled 'artifacts' which is a term that covers many undesirable effects. Movement artifacts stem from the subject moving during the scanning period, and can impose a serious degradation of image quality which is virtually impossible to correct post session. Some 3D acquisition sequences generate "phantom" replica of parts of the image volume superimposed on the volume. This is typically most predominant in one direction; by re-orienting slices accordingly, the "phantoms" can be placed outside the desired areas of interrest, e.g. the brain. RF 'glitches' can generate noisy pathces and lines across the image field of view. Field inhomogeneities in the scanner cause the intensity level of the scan to vary slowly across the image volume. This can be corrected for by fitting smooth functions to pixel samples of identical tissue types (obtained manually) like grey or white matter across the image volume. The image is corrected by rescaling the pixels accordingly, thus 'straightening' the smooth curve. To conclude, a good deal of image processing steps (and ditto 'hacks') are needed to go from the raw recorded MRI scan to an image stripped from skull and scalp and corrected for inhomogeneity, which can be segmented into tissue types like white matter, gray matter, CSF, as shown below in Figure 1.5.

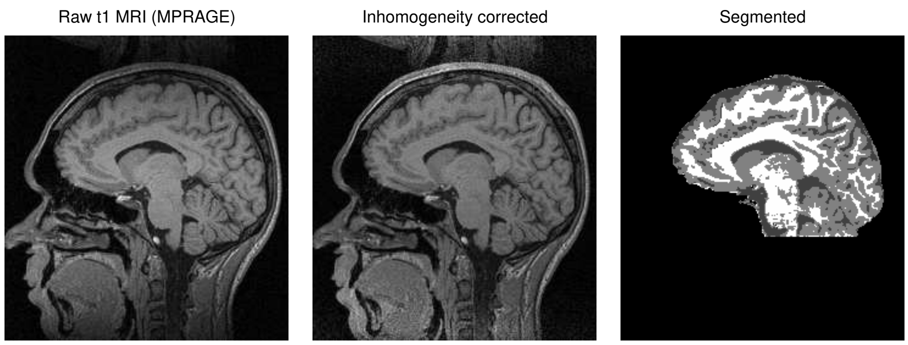


Figure 1.5: Raw t1 MRI (leftmost), intensity corrected (center) and stripped, segmented. The "phantom" artifact is visible in the unstripped data volumes, and is actually enhanced by the intensity correction.

Chapter 2

Bayesian Image Noise Reduction

The foundation of noise reduction is the statistical model of the noise and object being imaged. Noise reduction is a process which is often hindered by the absence a ground truth for direct evaluation of the restored image. This has led to the offspring of many image processing 'hacks', with the argument for introducing these methods being: If the method makes my end results look better, it must be doing something right. One main point of Chapter 3 is to take this argument for saying, utilizing the strength of the result of the statistical analysis of an activation study as the indicator for the quality of the preceding image processing steps. This evaluation strategy is directly applicable for the methods described in this section, although this work is yet to be performed.

Meanwhile, the Bayesian approach to the image restoration problem lets us combine our full prior knowledge with the observed data, lets us combine data from several sensors for sensor fusion and lets us compute not only the most probable value for the 'true' image but also the uncertainties associated with it. This does, however, rest on a firm belief in the statistical signal models used for the processing.

In the following we shall investigate methods for applying the Bayesian method on PET scans and we shall focus on model estimation and application. It will be investigated how the theory for stationary Gaussian signals (Wiener filters) can be extended to bounded signals, i.e. random signals that are bounded spatially by edges. The edge model is believed to be particularly useful for brain modeling since the brain anatomy enforces a boundary on the brain activation signal.

2.1 Bayes Relation for Images

Given an observed, degraded image $\boldsymbol{d}$ the aim is to reconstruct the underlying object signal[1] $\boldsymbol{u}$. The degradation is assumed to take the form

$$\boldsymbol{d} = \mathbf{H}\boldsymbol{u} + \boldsymbol{\varepsilon} \tag{2.1}$$

where $\mathbf{H}$ is a linear transformation matrix e.g. a blurring operation and $\boldsymbol{\varepsilon}$ is a noise process. The image is encoded as a single vector so that any convolution and/or image transformation (e.g. the Radon transform) may be included in $\mathbf{H}$. Using Bayes' relation it is possible to express the likelihood of an observed degraded image in factors of a distortion process and a prior process:

$$P(\boldsymbol{u} \,|\, \boldsymbol{d}, \mathbf{H}) = \frac{P(\boldsymbol{d} \,|\, \boldsymbol{u}, \mathbf{H}) P(\boldsymbol{u})}{P(\boldsymbol{d})} \propto P(\boldsymbol{d} \,|\, \boldsymbol{u}, \mathbf{H}) P(\boldsymbol{u}). \tag{2.2}$$

This states that if we are looking for the most likely $\boldsymbol{u}$ given an observed image $\boldsymbol{d}$, we need only maximize the product $P(\boldsymbol{d} \,|\, \boldsymbol{u}, \mathbf{H}) P(\boldsymbol{u})$

$$\boldsymbol{u}_{\text{MAP}} = \arg \max_{\boldsymbol{u}} P(\boldsymbol{d} \,|\, \boldsymbol{u}, \mathbf{H}) P(\boldsymbol{u}). \tag{2.3}$$

This is referred to as the Maximum A Posteriori (MAP) estimate. A difficulty in image restoration applications is the determination of suitable prior- and degeneration models and as well as the transformation

[1]The term 'signal' is used in place of 'image' for stressing that it is thought of as the output of a random process.

H. To maintain the data's credibility it is desirable to use consistent methods for choosing our model and parameters used for the image restoration.

The signal models considered in the following sections 2.2–2.3 are Gaussian random fields. In the stationary version these are well-understood and the signal properties can be determined using frequency domain techniques and the z-transform. In image processing stationarity is seldomly achieved—edge effects will not be negligible due to the limitation of the image boundaries. In brain imaging edge effects are predominant since the major part of brain activity resides in anatomically bounded areas of small size compared to the scanner's total field-of-view.

This is the motivation for investigating the bias effects caused by extending the theory of stationary random fields to the non-stationary case later in this chapter. We will also describe the us of Markov Random Fields as image models.

2.2 Signal Models for Stationary Gaussian Random Fields

A stationary Gaussian signal is a signal whose second order moments do not change with the image location. The first order moment is assumed to be zero, and since the Gaussian distribution is defined with its first and second moments, the signal is characterized from its covariance function. For three-dimensional volumes the auto-covariance function is defined as

$$r_u(i,j,k) = \langle u(p,q,r) - u(p+i,q+j,r+k) \rangle \tag{2.4}$$

The covariance function is directly related to the power spectrum density function

$$S_u(z_1,z_2,z_3) = \sum_{i,j,k=-\infty}^{\infty} r_u(i,j,k) z_1^{-i} z_2^{-j} z_3^{-k} \tag{2.5}$$

$$r_u(i,j,k) = \frac{1}{2\pi} \iiint_{\omega_1,\omega_2,\omega_3=-\pi}^{\pi} S_u(\omega_1,\omega_2,\omega_3) \exp\big[-j(\omega_1 i + \omega_2 j + \omega_3 k)\big]\, d\omega_1\, d\omega_2\, d\omega_3 \tag{2.6}$$

2.2.1 Linear Prediction

A given stationary Gaussian signal's power spectrum can be represented by a linear filter which is capable of predicting the signal itself using past or surrounding samples. The vector of filter coefficients is a compact representation of the power density of the signal; the prediction error has a much smaller entropy than the original signal, which is why these types of models are widely used for data-compression (Linear Predictive Coding, [1]). Different model architectures may be applied, generating Autoregressive Models (AR, all-pole models), Moving Average models (MA, all zero models), or both combined (ARMA), and the region of support may be chosen to yield non-, semi and causal impulse response models. We will only treat AR models here.

Auto Regressive Models

The AR prediction model for the three dimensional signal $u(i,j,k)$ is

$$\bar{u}(i,j,k) = \sum_{p,q,r\in\mathcal{W}} a(p,q,r) u(i-p, j-q, k-r) \tag{2.7}$$

which has an accompanying error sequence $\varepsilon(i,j,k) = u(i,j,k) - \bar{u}(i,j,k)$.

The region $\mathcal{W}$ (which does not include $(0,0,0)$) determines if the model is either causal, semi-causal or non-causal. A causal model is obtained if an ordering of voxels exists (e.g. row or column ordering) so that any voxel's prediction region contains only previous voxels. A semi-causal model is obtained if the model is causal in one or more directions and non-causal in one or more other directions. This is illustrated in the two dimension case in Figure 2.1. For three dimensional signals, the semi-causal model can be causal in one or two directions which will be referred to as the single- and double-causal model. Semi-causal models are useful for time series applications, e.g. fMRI data sets where time occupies the causal dimension and brain activity the non-causal dimensions. In the following the single-causal model is assumed to be causal in the k direction while the double-causal model is assumed causal in the j and k directions.

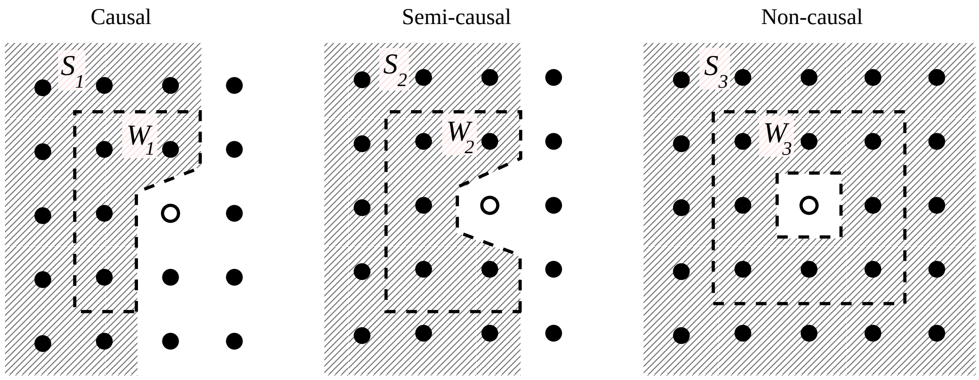


Figure 2.1: Prediction regions for the voxel marked '∘'. The shaded regions are the prediction regions $\mathcal{S}_{1..3}$ and the corresponding finite prediction regions $\mathcal{W}_{1..3} \subset \mathcal{S}_{1..3}$. For the finite causal prediction region $\mathcal{W}_1$ the central voxel depends only on previous voxels in a column ordering of the voxels. Note that (0,0) is not included in $\mathcal{S}$ and $\mathcal{W}$.

Minimum Variance Representation

The transfer function

$$A_p(z_1, z_2, z_3) = \varepsilon(z_1, z_2, z_3)/u(z_1, z_2, z_3) = 1 - \sum_{p,q,r \in \mathcal{W}} a(p,q,r) z_1^{-p} z_2^{-q} z_3^{-r} \tag{2.8}$$

defines the prediction error filter which has its name because the prediction error $\varepsilon(i,j,k)$ is obtained when the input is fed through this filter.

The filter coefficients are determined to minimize the variance of the prediction error (hence the name: Minimum Variance Representation) using the orthogonality condition

$$\forall (i,j,k) \in \mathcal{S}: \quad \big\langle \varepsilon(p,q,r) u(i+p, j+q, k+r) \big\rangle = \beta^2 \delta(i,j,k) \tag{2.9}$$

which states that the prediction error is uncorrelated with all voxel values in the prediction region. Inserting $\varepsilon = u - \bar{u}$ and using (2.7) and (2.4) we get

$$r_u(i,j,k) - \sum_{p,q,r \in \mathcal{W}} a(p,q,r) r_u(i-p, j-q, k-r) = \beta^2 \delta(i,j,k), \tag{2.10}$$

which gives an (infinite) set of equations to be solved for $(i,j,k) \in \mathcal{S}^+$.

For practical applications, the covariance function is truncated, solving (2.10) for a subset $\mathcal{Q} \subseteq \mathcal{S}^+$ e.g. $(i,j,k) \in \mathcal{W}^+$ where $\mathcal{W}^+ = \mathcal{W} \cup (0,0,0)$. When solving (2.10) for a larger area than $\mathcal{W}^+$ the system of equations is over-determined and β^2 may become ill-determined (and e.g. negative) if solved on equal terms as the remaining variables. To releave this first solve

$$\sum_{p,q,r \in \mathcal{W}} a(p,q,r) r_u(i-p, j-q, k-r) = r_u(i,j,k), \qquad (i,j,k) \in \mathcal{Q} \setminus (0,0,0) \subseteq \mathcal{S}, \tag{2.11}$$

and then estimate β^2 from

$$\beta^2 = r_u(0,0,0) - \sum_{p,q,r \in \mathcal{W}} a(p,q,r) r_u(-p,-q,-r). \tag{2.12}$$

The auto-covariance function may be difficult to estimate well enough, especially if there are long range correlations in the image. A conservative approach is to fit only the r_u's that are well-determined given the data.

The spectral density of the prediction error can be determined using (2.8) and (2.9) by first computing its covariance function

$$
\begin{aligned}
r_\varepsilon(i,j,k) &= \langle \varepsilon(p,q,r)\varepsilon(p-i,q-j,r-k)\rangle && (2.13)\\
&= \beta^2\Big[\delta(i,j,k) - \sum_{p,q,r\in\mathcal{W}} a(p,q,r)\delta(p+i,q+j,r+k)\Big] && (2.14)\\
&= \begin{cases}
\beta^2\delta(i,j,k) & \text{causal model,}\\
-\beta^2\delta(j,k)a(-i,0,0) \quad (-i,j,k)\in\mathcal{W}^+ & \text{double-causal model,}\\
-\beta^2\delta(k)a(-i,-j,0) \quad (-i,-j,k)\in\mathcal{W}^+ & \text{single-causal model,}\\
-\beta^2 a(-i,-j,-k) \quad (-i,-j,-k)\in\mathcal{W}^+ & \text{non-causal model}
\end{cases} && (2.15)
\end{aligned}
$$

where we have defined $a(0,0,0) = -1$. The resulting power spectrum is found by z-transformation of (2.15)

$$
S_\varepsilon(z_1,z_2,z_3) = \begin{cases}
\beta^2 & \text{causal model,}\\
\beta^2\sum_p a(p,0,0)z_1^{-p} = \beta^2 A_p(z_1,\infty,\infty) & \text{double-causal model,}\\
\beta^2\sum_{p,q} a(p,q,0)z_1^{-p}z_2^{-q} = \beta^2 A_p(z_1,z_2,\infty) & \text{single-causal model,}\\
\beta^2(1-\sum_{p,q,r\in\mathcal{W}} a(p,q,r)z_1^{-p}z_2^{-q}z_3^{-r}) = \beta^2 A_p(z_1,z_2,z_3) & \text{non-causal model,}
\end{cases} \quad (2.16)
$$

with the prediction error filter $A_p(z_1,z_2,z_3)$ defined in (2.8). Notice that only the causal model has an error signal with a white power spectrum. Using (2.8) we can express the signal power spectrum as

$$
S_u(z_1,z_2,z_3) = \begin{cases}
\dfrac{\beta^2}{A_p(z_1,z_2,z_3)A_p(z_1^{-1},z_2^{-1},z_3^{-1})} & \text{causal model,}\\[2ex]
\dfrac{\beta^2 A_p(z_1,\infty,\infty)}{A_p(z_1,z_2,z_3)A_p(z_1^{-1},z_2^{-1},z_3^{-1})} & \text{double-causal model,}\\[2ex]
\dfrac{\beta^2 A_p(z_1,z_2,\infty)}{A_p(z_1,z_2,z_3)A_p(z_1^{-1},z_2^{-1},z_3^{-1})} & \text{single-causal model,}\\[2ex]
\dfrac{\beta^2}{A_p(z_1,z_2,z_3)} & \text{non-causal model,}
\end{cases} \quad (2.17)
$$

Filter Stability and Spectral Match

There is no guarantee that the fitted model will be stable so that the inversions in (2.17) can actually be performed. Experience shows that the stability of the filter tends to improve with increasing filter size. Increasing the filter size means that more and more of the auto-covariance values will be matched and the model spectrum will approximate the signal spectrum as the model size approaches infinity, so that stability eventually is reached.

The AR model is an all-pole model and filter instability is a problem tied to this representation of the signal spectrum. To investigate the filter instability problem further we express the prediction error variance in the frequency domain [4]. Letting the signal spectrum be denoted by S_u^* and using $\varepsilon(z_1,z_2,z_3) = u^*(z_1,z_2,z_3)A_p(z_1,z_2,z_3)$ and Parsevals relation we have

$$
\begin{aligned}
\langle\varepsilon^2\rangle &= \frac{1}{8\pi^3}\iiint_{-\pi}^{\pi} \varepsilon(z_1,z_2,z_3)\varepsilon(z_1^{-1},z_2^{-1},z_3^{-1})\,d\omega_1\,d\omega_2\,d\omega_3 && (2.18)\\
&= \frac{1}{8\pi^3}\iiint_{-\pi}^{\pi} S_u^* A_p(z_1,z_2,z_3)A_p(z_1^{-1},z_2^{-1},z_3^{-1})\,d\omega_1\,d\omega_2\,d\omega_3.
\end{aligned}
$$

Inserting the model spectrum for non-, single-, double- and causal models we obtain

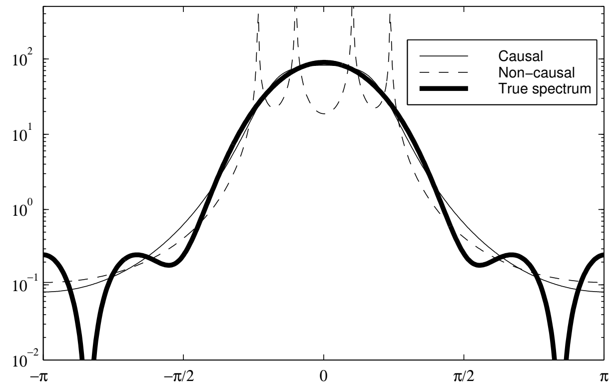
$$\langle \varepsilon^2(i,j,k) \rangle = \begin{cases} \frac{\beta^2}{8\pi^3} \iiint_{-\pi}^{\pi} \frac{S_u^*(z_1,z_2,z_3)}{S_u(z_1,z_2,z_3)}\, d\omega_1\, d\omega_2\, d\omega_3 & \text{causal model,} \\ \frac{\beta^2}{8\pi^3} \iiint_{-\pi}^{\pi} \frac{S_u^*(z_1,z_2,z_3)}{S_u(z_1,z_2,z_3)} A_p(z_1,\infty,\infty)\, d\omega_1\, d\omega_2\, d\omega_3 & \text{double-causal model,} \\ \frac{\beta^2}{8\pi^3} \iiint_{-\pi}^{\pi} \frac{S_u^*(z_1,z_2,z_3)}{S_u(z_1,z_2,z_3)} A_p(z_1,z_2,\infty)\, d\omega_1\, d\omega_2\, d\omega_3 & \text{single-causal model,} \\ \frac{\beta^2}{8\pi^3} \iiint_{-\pi}^{\pi} \frac{S_u^*(z_1,z_2,z_3)}{S_u(z_1,z_2,z_3)} A_p(z_1,z_2,z_3)\, d\omega_1\, d\omega_2\, d\omega_3 & \text{non-causal model.} \end{cases} \tag{2.19}$$

Therefore we see that minimizing the squared prediction error is equivalent to minimizing the integrated ratio of the signal spectrum S_u^* to the model approximation S_u. For the acausal (i.e. all but causal) models the integral is further weighted with a factor proportional to the prediction error power spectrum. Unstable filters for which $S_u \to \infty$ are not heavily penalized by this error measure; this is the reason for the stability problems that typically arise using AR models. For the non-causal model, the problem is even worse since the weighting factor is proportional to $A_p(z_1, z_2, z_3)$ which is closer to zero at the model's pole frequencies.

Being an all-pole model, the AR model is well suited for approximating spectra with peaks and has a tendency for having difficulties representing a slow-varying power spectrum, for example variations caused by low-order spectral zeroes. Any peaks in the signal spectrum can be expected to be matched well while an equally poor match is obtained in between the peaks.

Another interpretation of (2.19) is that the contributions to the error integral for regions of the power-spectrum where $S_u < S_u^*$ is canceled by regions where $S_u > S_u^*$. This suggests the application of other error measures in place of the least square approach, such as $E = \int \left[\log \mathcal{S}(\omega) - \log \mathcal{S}^*(\omega)\right]^2 d\omega$. The disadvantage of such error measures is that the parameter estimation problem becomes non-linear, which means that iterative and thus computationally more demanding methods are necessary. We will not treat such alternative methods here although they are worth investigating.

Example 2.1 *In this example, we will fit a causal and a non-causal one-dimensional AR model to synthetic data. The data was obtained by applying the FIR filter sequence* $[0.5, 1, 2, 2.5, 2, 1, 0.5]$ *to a Gaussian white noise sequence. Note that this generates an all-zero spectrum. The AR model parameters were computed by solving (2.11) for* $(i,j,k) \in \mathcal{W}_1$ *and* $\mathcal{W}_3$ *respectively. The figure below shows the true power spectrum compared to the fitted causal and non-causal models. It is evident that the non-causal model applies a much more peaked spectrum, in fact the fitted non-causal model is not stable. The causal model shows a more desirable behavior, although these models can become unstable too.*



■

Creating Non-causal models from Causal Models

From (2.17) we see that, given a causal AR model, we can readily create a non-causal model by solving $\mathcal{S}_{noncau}(z_1, z_2, z_3) = \mathcal{S}_{cau}(z_1, z_2, z_3)$

$$\begin{aligned} \frac{\beta^2_{noncau}}{A_{p,noncau}(z_1, z_2, z_3)} &= \frac{\beta^2_{cau}}{A_{p,cau}(z_1, z_2, z_3)} \cdot \frac{\beta^2_{cau}}{A_{p,cau}(z_1^{-1}, z_2^{-1}, z_3^{-1})} \\ A_{p,noncau}(0,0,0) &= 1. \end{aligned} \tag{2.20}$$

This approach may releave some of the stability problems of the non-causal models discussed above: Instead of solving (2.11, 2.12) for the non-causal model, we first build the causal model and translate into non-causal representation. With this approach, the non-causal model will only become unstable when the causal model does so.

Separable Models

The computational problem of estimating the AR model parameters and actually applying the recursive filters is non-trivial, especially for the acausal filters since the filter operation needs to be performed repeatedly until the signal converges. A separable filter is much faster to apply, since the multidimensional filter operation can be replaced with sequential 1-dimensional filter operations.

The separable model assumes that the auto-covariance function can be separated so that, e.g.

$$r_u(i,j,k) = r_1(i) r_2(j,k). \tag{2.21}$$

When fitting $S_1(z_1)$ to $r_1(i)$ and $S_2(z_2, z_3)$ to $r_2(j,k)$ the signal spectrum is expressed as

$$S(z_1, z_2, z_3) = S_1(z_1) S_2(z_2, z_3). \tag{2.22}$$

Example 2.2 *In this example we will fit two-dimensional AR models to the PET background noise. Two* $\left[{}^{15}O\right]$ *water PET scans with subjects in a resting condition were acquired as part of an activation study. The voxel size was* $2 \times 2 \times 4.25$ *mm and the scans were co-registered by a rigid body transformation (6 parameter AIR, [5, 6]) and re-sampled using sinc interpolation. Subtracting the two slices shown in Figure 2.2 results in an estimate of the PET reconstruction noise disregarding differences in cerebral blood flow between the two scans. The noise auto-covariance function is estimated inside the area marked with the rectangle and is used for spectral analysis.*

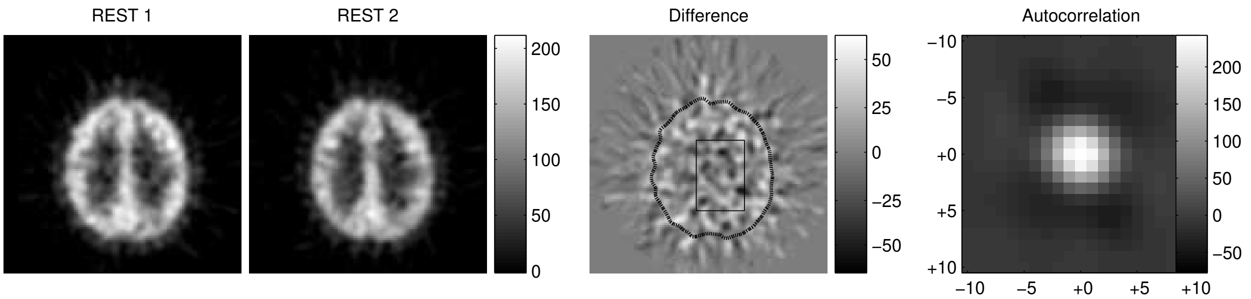


Figure 2.2: PET background noise is estimated by subtracting two scans acquired with the subject resting. The auto-covariance function is computed in the area inside the rectangle.

Non-, semi-causal and causal models ($\mathcal{W}_{1..3} = \{(i,j) \in S_{1..3} | -5 \leq i \leq 5 \cap -5 \leq j \leq 5\}$) *were fitted by solving (2.11) and (2.12) for* $(i,j,k) \in \mathcal{Q}$ *which covered all* 21×21 *estimated values of the auto-covariance function. This yielded over-determined systems of equations which were solved using the least squares method. These solutions are shown as the three leftmost prediction error filters in Figure 2.3. The causal filter was translated to the non-causal representation using (2.20) shown as the fourth filter in this Figure. Finally, one-dimensional causal filters of order 11 were fitted to* $r_u(0,j), j = 1..11$ *and* $r_u(i,0), i = 1..11$ *of the auto-covariance function. These*

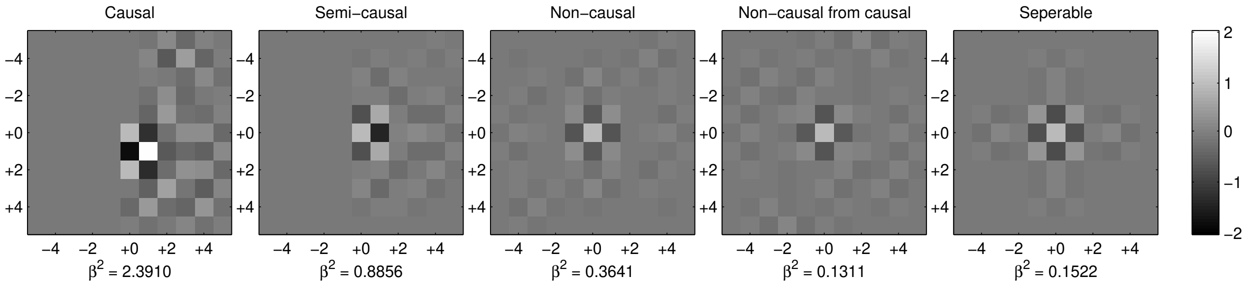


Figure 2.3: 11-by-11 prediction error filters A_p for a) causal, b) semi-causal, c) non-causal models, d) the filter obtained by creating a non-causal model from the causal model using (2.20) and e) The 2D filter obtained by fitting 2 1D causal filters.

filters were translated into two-dimensional non-causal representation using (2.20) and (2.22), shown as the fifth prediction error filter in Figure 2.3.

The fitted non-causal filter (third shown) is unstable—the two-dimensional power spectrum for the prediction error contains zero frequencies so that the signal spectrum cannot be expressed using (2.17). This shows in the image representation of the power spectrum in Figure 2.4 as white 'spots' in the numerically computed spectrum.

We see that the causal model (and the derived non-causal representation) seems to yield the most stable signal representation. It should be mentioned that other experiments have yielded unstable causal filters as well. The separable filter estimation procedure is the most reliable, especially when each low-dimensional match is performed as a causal filter. To conclude, this example demonstrates that the filter stability problem is worse when estimating multidimensional spectra compared to the one-dimensional case, and the application of causal filters seems most stable.

■

2.2.2 Converting AR Models to Signal Probability Models

The AR model is closely related to the covariance matrix of the signal. This symmetrical and positive definite matrix is defined as

$$\Sigma_{\mathbf{u}} = \langle \boldsymbol{u}\boldsymbol{u}^T \rangle \tag{2.23}$$

for the signal encoded in the vector $\boldsymbol{u}$. Signal stationarity implies that the signal has infinite length, making $\Sigma_{\mathbf{u}}$ of infinite size. Furthermore, the matrix becomes Toeplitz[2] and the auto-covariance function is displayed in the columns (or rows) of the covariance matrix. For a multidimensional signal encoded in the vector $\boldsymbol{u}$, the covariance matrix contains multiple bands—although they are infinitely separated.

We may also note that, multiplying two (infinite) Toeplitz matrices corresponds to the convolution of their column vectors, for example $[1,2,1] * [2,1,2] = [2,5,6,5,2]$ as shown below.

$$\begin{bmatrix} \ddots & \ddots & & & \\ \ddots & 2 & 1 & & \\ & 1 & 2 & 1 & \\ & & 1 & 2 & \ddots \\ & & & \ddots & \ddots \end{bmatrix} \cdot \begin{bmatrix} \ddots & \ddots & & & \\ \ddots & 1 & 2 & & \\ & 2 & 1 & 2 & \\ & & 2 & 1 & \ddots \\ & & & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} \ddots & \ddots & \ddots & & & & \\ \ddots & 6 & 5 & 2 & & & \\ \ddots & 5 & 6 & 5 & 2 & & \\ & 2 & 5 & 6 & 5 & 2 & \\ & & 2 & 5 & 6 & 5 & \ddots \\ & & & 2 & 5 & 6 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.24}$$

Let $R_u(i,j,k)$ determine the columns of $\Sigma_{\mathbf{u}}$. Now the columns of the inverse $\Sigma_{\mathbf{u}}^{-1}$ (which is also symmetrical and Toeplitz) define the inverse covariance $R_u^{-}(i,j,k)$ where the identity

$$R_u(i,j,k) \; * \; R_u^{-}(i,j,k) = \delta(i,j,k) \tag{2.25}$$

[2] A Toeplitz matrix is a matrix whose (sub)diagonal elements are constant along the (sub)diagonal.

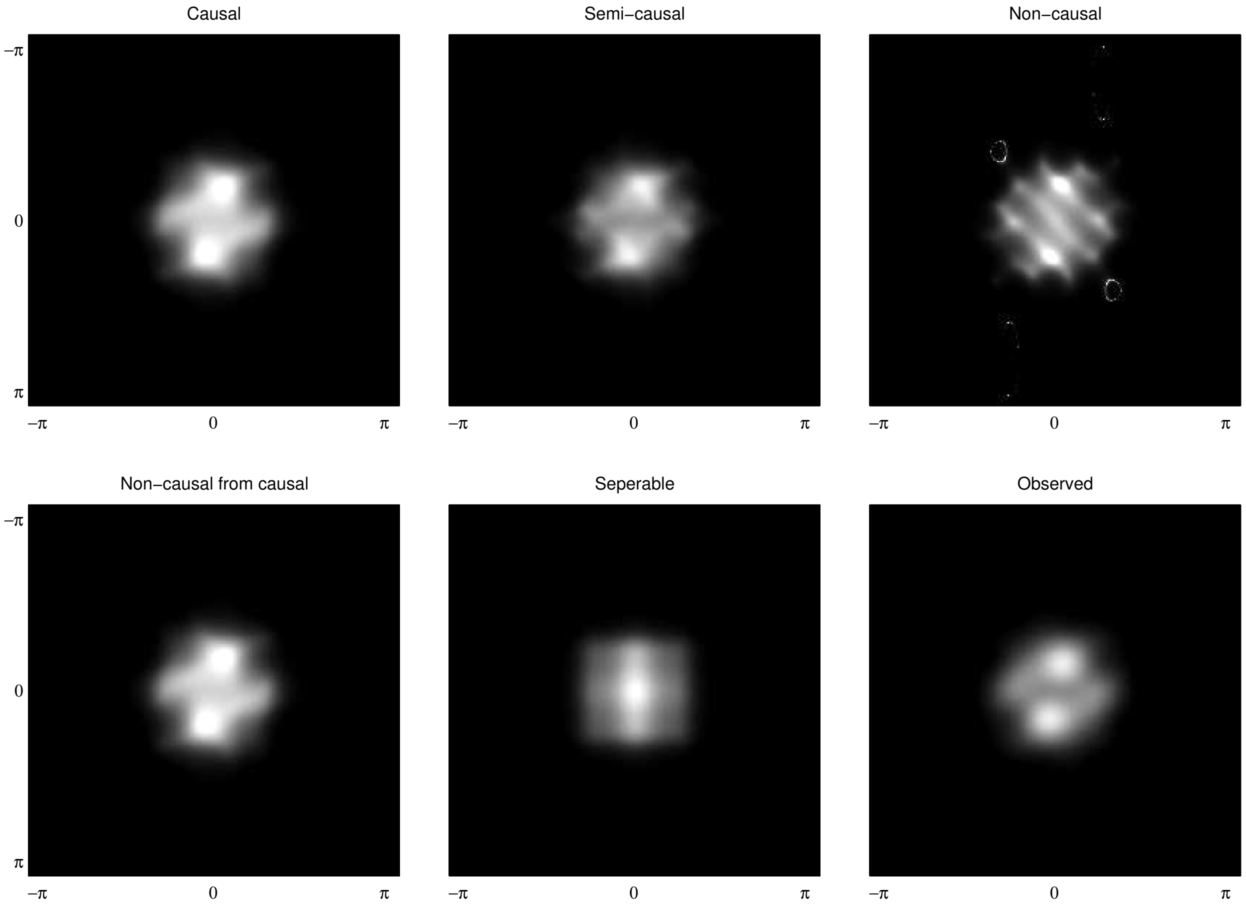


Figure 2.4: Estimated signal power spectra of fitted AR models. The observed spectrum is computed by Fourier transformation of the auto-covariance function padded with zeros.

must apply, given that $\Sigma_{\mathbf{u}}^{-1}\Sigma_{\mathbf{u}} = \mathbf{I}$ and using the convolution analogy. By Fourier transforming (2.25) $R_u \leftrightarrow S_u$ we get $S_u(z_1, z_2, z_3) = 1/S_u^{-}(z_1, z_2, z_3)$. Comparing this with the expression for the power spectrum of the non-causal AR model (2.17) we see that $S_u^{-}(z_1, z_2, z_3) = A_p(z_1, z_2, z_3)/\beta^2$. Inverse Fourier transforming this relation is simple given (2.8): The elements of $R_u^{-}(i, j, k)$ that enter the columns of the inverse covariance matrix are simply the parameters $a(p, q, r)$ divided by the prediction error variance β.

$$R_u^{-}(p, q, r) = -\frac{a(p, q, r)}{\beta^2}, \qquad\qquad R_u^{-}(0, 0, 0) = \frac{1}{\beta^2} \tag{2.26}$$

In other words, we can readily transform the non-causal AR model into an inverse covariance matrix for the multivariate Gaussian distribution, regarding the observed signal as a single multivariate stochastic variable

$$P(\boldsymbol{u}) = \sqrt{\frac{|\Sigma_{\mathbf{u}}^{-1}|}{(2\pi)^N}} \exp\Big[-\frac{1}{2}\boldsymbol{u}^T \Sigma_{\mathbf{u}}^{-1} \boldsymbol{u}\Big]. \tag{2.27}$$

This distribution is of infinite dimensionality for for signal of infinite length. Practical signals are most often truncated in length which introduces a boundary problem. The boundary problem is typically overcome by the application of windowing functions, of which many different exist: Hanning, Hamming, Lanczos, Bartlett, Gaussian etc., see e.g. [7]. Representing the signal covariances using the multivariate distribution we obtain full control over the model at the boundaries of the signal. We shall investigate this further in Section 2.4.

2.3 The Wiener Filter

The theory of Wiener filters is a special case of the Bayesian restoration using prior models. The Wiener filter theory applies to Gaussian signals, attempting to recover a signal which has been degraded by a linear transformation (for example a convolution) and additive noise. The recovery is based on knowledge about the transformation (blurring kernel) and the covariance properties of the noise and the unknown object signals.

In matrix notation, the degeneration assumes the general form

$$\boldsymbol{d} = \mathbf{H}\boldsymbol{u} + \boldsymbol{n}, \tag{2.28}$$

where $\boldsymbol{d}$, $\boldsymbol{u}$ and $\boldsymbol{n}$ are zero mean Gaussian processes, and $\mathbf{H}$ is a known transformation matrix. If all signals are assumed stationary, the degeneration can be expressed in terms of convolutions

$$d(p,q,r) = u(p,q,r) * h(i,j,k) + n(p,q,r), \tag{2.29}$$

We wish to restore $u(p,q,r)$ by designing an optimal filter $g(i,j,k)$ which undoes the above degeneration by means of a convolution

$$\hat{u}(p,q,r) = u(p,q,r) * g(i,j,k) \tag{2.30}$$

while minimizing the squared error

$$\left\langle \big(\hat{u}(p,q,r) - u(p,q,r)\big)^2 \right\rangle \tag{2.31}$$

The filter $g(p,q,r)$ which applies to stationary signals is called the Fourier Wiener Filter. More generally, we can specify the Wiener filter as the matrix $\mathbf{G}$ so that $\hat{\boldsymbol{u}} = \mathbf{G}\boldsymbol{d}$.

The Fourier Wiener Filter

The filter is identified by applying the orthogonality relation, which states that the error $u - \hat{u}$ and the observed signal d are uncorrelated processes

$$\big\langle (u(i,j,k) - \hat{u}(i,j,k)) d(i',j',k') \big\rangle = 0, \qquad \forall (i,i',j,j',k,k') \tag{2.32}$$

Inserting (2.29) yields

$$\big(u(i,j,k) - \big(u(i,j,k) * h(p,q,r) + n(i,j,k)\big) * g(p,q,r)\big) \star \big(u(i,j,k) * h(p,q,r) + n(i,j,k)\big) = 0 \tag{2.33}$$

where we have used the symbol '$\star$' to represent the correlation operator. By Fourier transformation (for brevity writing $U = U(z_1, z_2, z_3)$) we obtain

$$(U - (UH + N)G) \cdot \overline{(UH + N)} = 0 \tag{2.34}$$

Using that the signals $u(i,j,k)$ and $n(i,j,k)$ are independent so that $UN = 0$, and further solving for G we obtain the expression for the Fourier Wiener filter

$$G = \frac{S_u \overline{H}}{S_u H \overline{H} + S_n} \tag{2.35}$$

The filter coefficients are recovered by inverse transformation of the above equation. The filter obtained will, in general, be of infinite size and will need to be truncated by means of a windowing function or similar technique. It is also possible to design finite length Wiener filters directly, see e.g. [1, Ch. 8.4].

The General Wiener Filter

Without the stationarity constraint, the Wiener Filter becomes a matrix transformation, which recovers the linear transformation $\mathbf{H}$ taking account of the noise statistics. We may derive the Wiener filter matrix from the Bayesian relation (2.2)

$$P(\boldsymbol{u} \,|\, \boldsymbol{d}) \propto P(\boldsymbol{d} \,|\, \boldsymbol{u}) P(\boldsymbol{u}) \propto \exp\Big[-\frac{1}{2}(\mathbf{H}\boldsymbol{u} - \boldsymbol{d})^T \Sigma_N^{-1} (\mathbf{H}\boldsymbol{u} - \boldsymbol{d}) - \frac{1}{2}\boldsymbol{u}^T \Sigma_u^{-1} \boldsymbol{u} \Big] \tag{2.36}$$

where $\Sigma_u = \langle \boldsymbol{u}\boldsymbol{u}^T \rangle$ and $\Sigma_N = \langle \boldsymbol{n}\boldsymbol{n}^T \rangle$. Minimizing with respect to $\boldsymbol{u}$ yields

$$\left(\mathbf{H}^T \Sigma_N^{-1} \mathbf{H} + \Sigma_u^{-1}\right)\boldsymbol{u} = \mathbf{H}^T \Sigma_N^{-1} \boldsymbol{d} \tag{2.37}$$

Provided the inverse exists, we thus have

$$\mathbf{G} = \left(\mathbf{H}^T \Sigma_N^{-1} \mathbf{H} + \Sigma_u^{-1}\right)^{-1} \mathbf{H}^T \Sigma_N^{-1} \tag{2.38}$$

We note the following

1. When the covariance matrices and the transformation matrix have Toeplitz structure, we can interpret matrix multiplication as convolution, and the transpose operator as reversal in time for the appropriate discrete signals. In the Fourier domain, this translates again into spectral multiplication and complex conjugation, respectively. By application of these 'rules' we readily see that (2.35) follows from (2.38).
2. In the absence of noise, taking the limit $\Sigma_N^{-1} = \mathbf{Y}/\sigma^2$, $\sigma \to 0$, the Wiener filter matrix is identical to the Moore-Penrose inverse.

$$\mathbf{G} = \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \mathbf{H}^T, \tag{2.39}$$

3. With no transformation matrix, the recovering matrix reads

$$\mathbf{G} = \left(\Sigma_N^{-1} + \Sigma_u^{-1}\right)^{-1} \Sigma_N^{-1} = \left(\Sigma_N + \Sigma_u\right)^{-1} \Sigma_u. \tag{2.40}$$

4. Inversion of a symmetrical positive definite matrix is best performed using Cholesky factorization, which can be performed 'in place' and is a factor 2 faster than LU-decomposition (see e.g. [8]).

2.4 Markov Random Field Models

The Markov Random Field is a class of statistical models for random variables that possesses certain properties which make them well suited for image models. A Markov field consists of a collection or a lattice of random variables (pixel values) with local interaction. The Markov property dictates that the conditional distribution of any variable in the random field given the remaining field is identical to the distribution conditioned only on values in a finite size local clique of neighboring pixel values

$$P(u_i \,|\, u_j, j \neq i) = P(u_i \,|\, u_j, j \in \mathcal{C}_i) \qquad\qquad i \notin \mathcal{C}_i. \tag{2.41}$$

By limiting the size of the cliques $\mathcal{C}_i$ the Markov model achieves the desirable property that the computational load of computing (2.41) for the entire field of variables $i = 1..N$ scales with $\mathcal{O}(N)$. It is useful to express the simultaneous distribution of all variables in Gibbs form

$$P(\boldsymbol{u}) = Z^{-1} \exp\left(-E(\boldsymbol{u})\right), \tag{2.42}$$

where the energy function $E(\boldsymbol{u})$ is bounded from below. The normalization constant Z (aka. the partition function) is determined so that $P(\boldsymbol{u})$ is a density function

$$Z = \int_{\boldsymbol{u}} \exp\left(-E(\boldsymbol{u})\right) d\boldsymbol{u}. \tag{2.43}$$

and is therefore also a function of parameters entering the energy function. Recalling the Bayesian observation model's expression for the posterior distribution given the observed signal $\boldsymbol{d}$

$$P(\boldsymbol{u} \,|\, \boldsymbol{d}) = \frac{P(\boldsymbol{d} \,|\, \boldsymbol{u}) P(\boldsymbol{u})}{P(\boldsymbol{d})} \propto P(\boldsymbol{d} \,|\, \boldsymbol{u}) P(\boldsymbol{u}), \tag{2.44}$$

we may express this distribution by assigning Gibbs distributions to the degeneration process $E_{degen}(\boldsymbol{d}, \boldsymbol{u}) = P(\boldsymbol{d} \,|\, \boldsymbol{u})$ and the object prior model $E_{prior}(\boldsymbol{u}) = P(\boldsymbol{u})$

$$-\log P(\boldsymbol{u} \,|\, \boldsymbol{d}) = E_{degen}(\boldsymbol{d}, \boldsymbol{u}) + E_{prior}(\boldsymbol{u}) + \text{const.} \tag{2.45}$$

and the Maximum A Posteriori estimate (MAP) minimizes the cost (or energy) function

$$\boldsymbol{u}_{\mathrm{MAP}} = \arg \overset{\max}{\boldsymbol{u}} E(\boldsymbol{u}) = \arg \overset{\max}{\boldsymbol{u}} \left[E_{degen}(\boldsymbol{d}, \boldsymbol{u}) + E_{prior}(\boldsymbol{u})\right]. \tag{2.46}$$

Hence, all we need is a *forward* model of the degeneration process and an *object model* in order to reverse the degeneration. In this way the problem of maximization of the posterior probability is now expressed as the problem of minimizing a cost function. We may employ different optimization schemes for solving this problem, depending on the nature of the energy functions. When the energy function is a quadratic form, the solution of the above equation can be expressed as a least squares problem solvable by matrix inversion; the linear system of equations then become identical to the General Wiener filter equations (2.38).

We will Discuss how to design the energy functions, before we take a look at annealing based methods for energy function minimization.

2.4.1 Energy Function Selection

The design of the energy function calls for some tailoring to the specific restoration problem at hand. The degeneration model is best constructed from prior knowledge of the imaging system and the object being imaged. Signal de-convolution can be achieved in this manner. For example, Sørensen et al. [9] de-convolve the haemodynamic response function in fMRI images of the human brain using MRF signal degeneration models.

We need to decide whether the object should be represented by a discrete or continuous field. The above general optimization problem is typically solvable using both representations; a discretization of the object state space can induce important prior information which acts as a regularizer so that the restoration problem is better posed. In some imaging applications, one is highly interested in accurately estimating the intensity level; in this case a discretization may cause us to loose valuable information, leading us to apply continuous state variable models. It is also possible to create *compound* models having both discrete and continuous state variables.

The Boltzmann Machine is an example of a system of binary state variables with interactions. The state variables are divided into a hidden and visible part, corresponding to a binary observed vector $\boldsymbol{d}$ and an unknown binary object $\boldsymbol{u}$. Collecting the state vector in $\boldsymbol{x} = (\boldsymbol{u}, \boldsymbol{d})$ the Boltzmann energy function reads

$$E_{\mathrm{BM}}(\boldsymbol{x}) = \sum_{i,j} x_i x_j w_{i,j} = \boldsymbol{x}^T \mathbf{W}^{-1} \boldsymbol{x}, \qquad \mathbf{W}_{i,j} = w_{i,j} \tag{2.47}$$

where we see that the $w_{i,j}$ may be partitioned so that we obtain the form similar to (2.45) $E_{\mathrm{BM}} = E_{prior}(\boldsymbol{u}; \boldsymbol{w}) + E_{degen}(\boldsymbol{d}, \boldsymbol{u}; \boldsymbol{w})$.

By clamping the visible state elements to the observed values and letting the remaining hidden variables fluctuate according to the energy function, the Boltzmann Machine is able to generate samples of the distribution of the hidden variable given the observed clamped state. This is interpreted as an *associative memory*, or alternatively as performing *pattern completion* [10] and has attracted considerable interest for the understanding of the functionality of large 'wet' neuronal systems like the human brain, see e.g. [11].

The Boltzmann Machine is but one of many possible models based on the Gibbs family of state distributions. In Section 2.5.1, we shall focus on the interpretation of clamping the observed state variable for obtaining the distribution of the hidden variable. By adopting this framework in our image restoration setting (i.e. thinking of the hidden state variable as the object we are looking for), we obtain a means of acquiring the statistics of the restored image given the observed degraded image: We 'clamp' the visible state variables (the observed image) in the energy function and simulate the distribution of the hidden units according to the energy function, thus allowing us not only to obtain the most probable restored image, but also obtain error bars by simply measuring the appropriate statistics of the generated samples. This approach is called inference by Markov Chain Monte Carlo methods; a comprehensive presentation of this class of methods is given by Neal in [12].

There is, however, one more important reason for generalizing the framework of the Boltzmann Machine to image restoration, namely the useful result of Hinton and Sejnowski [13] that the weights (i.e. parameters) of the Boltzmann Machine can be inferred by a gradient method labelled the Boltzmann Machine Learning Rule. We shall later investigate a generalization of the Boltzmann Learning rule to the entire class of Gibbs distributions in Section 2.5.3 which was first introduced by Andersen and Hansen [15]. Training of log-transformed parameters using the generalized Boltzmann is discussed in Section 2.5.4.

Energy Functions for Image Restoration

A very common type of image prior models are based on the squared difference of neighboring pixel sites which in two dimensions reads

$$E_{prior}(\boldsymbol{u}) = \frac{\nu}{2}\sum_{i,j}\sum_{p,q} M_{p,q}(u_{ij} - u_{i+p,j+q})^2 \tag{2.48}$$

where $M_{p,q}$ describe the local object connectivity defined over some region of support[3] $(p,q) \in \mathcal{W}$. The covariance matrix of the AR model can be expressed similarly, since the above equation can be written in matrix form

$$E_{prior}(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T \Sigma_u^{-1} \boldsymbol{u} \tag{2.49}$$

where Σ_u^{-1} is a banded Toeplitz and symmetrical matrix

$$\Sigma_u^{-1} = \frac{\nu}{2}\begin{bmatrix} \ddots & & & & & & \vdots & & & & & \\ & & & & & & -2M_{2,0} & & & & & \\ & & \ddots & & & & \vdots & & & & & \\ & & & & & & -2M_{1,0} & & & & & \\ & & & & \ddots & & \vdots & & & & & \\ -2M_{1,1} & \cdot & 0 & \cdot & 0 & \cdot\ -2M_{1,0} & 4\sum\limits_{pq>0} M_{pq} + M_{0,0} & -2M_{1,0} & \cdot\ 0 & \cdot\ 0 & -2M_{1,1} \\ & & & & & & -2M_{1,0} & \ddots & & & \\ & & & & & & -2M_{2,0} & & \ddots & & \\ & & & & & & \vdots & & & & \ddots \end{bmatrix} \tag{2.50}$$

In the case of continuous variables, we can compare this expression with (2.27) to see that

$$\frac{1}{\beta^2} = \nu(2\Sigma M_{pq} - M_{0,0}), \qquad \frac{a_{pq}}{\beta^2} = -2M_{pq}, \tag{2.51}$$

which means that the AR non-causal model estimation techniques can be used for estimating the parameters M_{pq} provided all signals are stationary. As described in previous Section 2.2.1, the AR model is an all-pole model, which means the general form of the energy function (2.48) has a built in stability problem. We shall later discuss methods for sidestepping these problems. A simple way to ensure stability is, for example, to restrict the parameters M_{pq} to be strictly positive. This forces the object model to be a smoothing prior (favoring slow varying functions), which may or may not be desirable depending on the specific application.

Some workers have argued that the squared prior models (2.48) penalizes edges (i.e. sharp boundaries) to hard and have investigated methods for reducing the penalty for discontinuities in the image. One example is to use

$$E_{prior}(\boldsymbol{u}) = \frac{\nu}{2}\sum_{i,j}\sum_{p,q} M_{p,q}\log\cosh(u_{ij} - u_{i+p,j+q}) \tag{2.52}$$

which has linear limiting behavior and is used by Green [16]. Another possibility is to reduce the exponential yielding the 'power model' used by Higdon et al. [17]

$$E_{prior}(\boldsymbol{u}) = \frac{\nu}{2}\sum_{i,j}\sum_{p,q} M_{p,q}|u_{ij} - u_{i+p,j+q}|^d \tag{2.53}$$

where d is between one and two. Another approach, initially investigated by Geeman et al. [18], is to introduce an auxiliary edge map, defined by the line-sites $p_{i,j}$ and $q_{i,j}$. The function of the line sites is to

[3] Whether the region of support $\mathcal{W}$ is chosen to be causal or not does not alter the model. Writing (2.48) in the form $E_{prior}(\boldsymbol{u}) = \sum_{i,j}\Sigma_{p,q}\omega_{i,j;p,q}u_{i,j}u_{p,q}$ we need $\omega_{i,j;p,q} = \omega_{p,q;i,j}$ in order for an energy function to exist ([10]), which is identical to the requirement that Σ_u of (2.49) must be symmetrical. A non-causal $\mathcal{W}$ will have $M_{\mathrm{noncau},p,q} = M_{\mathrm{noncau},-p,-q} = 1/2M_{\mathrm{causal},p,q}$ for $p \geq 0, q \geq 0$. This requires twice the multiplications in the evaluation of (2.49), which is why $\mathcal{W}$ typically is chosen as a causal region with $M_{-p,-q} = 0$.

decouple the values of neighboring pixel values. When M_{pq} consists of nearest neighbor connectivity only, the line site model applied to the squared penalty function reads

$$E_{prior}(\boldsymbol{u}) = \frac{\nu}{2} \sum_{i,j} \left[(u_{ij} - u_{i+1,j})^2 \, p_{i,j} + (u_{ij} - u_{i,j+1})^2 \, q_{i,j} \right] \tag{2.54}$$

This is more complex to implement when M_{pq} has a large region of support, see e.g. [19] and [20]. An example of an application of the line-site model is given in Example 2.3.

The above mentioned models is but a few of the host of possibilities which are applicable to the image modeling problem. We shall shortly address the problem of model selection in the Bayesian Analysis, while focusing on Maximum likelihood estimation of parameters in the above type of image models.

2.4.2 Restoration by Annealing

A particular group of methods that has attracted attention for solving combinatorial optimization problems is the simulated annealing type of algorithms, which is a statistical mechanics formulation of the optimization problem. Annealing methods were introduced in connection with image processing by Geeman et al. [18].

The main reason for the popularity of this type of methods is the result that simulated annealing is capable of—at least at the limit of infinite computing time—localizing the mode(s) of the energy function even in the presence of local minima. The theoretical result applies only to discrete optimization problems, though. Geeman et al. [18] proves that there exists a T_1 for which the temperature scheme $T_t = T_1/\log(t)$, $t = 2, 3, \ldots, \infty$, as $T \to 0$ will make the simulated annealing procedure find the global minimum configuration. Within finite computation time, we cannot be sure that the result of the annealing procedure *is* the global minimum. Annealing methods have earned reputation by, among other things, (almost) solving the traveling salesman problem and other combinatorial problems, see e.g. [8, 21, 22]. The 'annealing' is an analogy with thermodynamics, with the way liquids freeze or crystallize. The atoms are able to align themselves up in a way to minimize the total energy of the system when the crystal is cooled, so that the total energy reaches the global minimum state out of all possible configurations of the atoms. The numerical simulation of this process, simulated annealing, is the process of simulating the thermal fluctuations of the system at the imaginary temperature T and gradually lowering T according some annealing schedule, until the system eventually settles in a minimum state. A temperature is introduced by scaling the energy function by $\beta = 1/T$, which corresponds to raising the probability distribution to the $1/T$'th power and re-normalizing so the distribution sums to unity. This operation does not change the location of the global maximum, because the locations of local extrema of the function $Z^{-1}(\beta)\exp(-\beta E)$ do not depend on β as long as $\beta > 0$. At high temperatures (β small) the density $Z^{-1}(\beta)\exp(-\beta E)$ becomes smooth, assigning almost equal probabilities to all states of the system. Opposite at low temperatures, more and more of the mass of the probability function is centered around the global maximum thereby forcing the system into this state.

It is possible to generate samples from a Gibbs energy function by generating a Markov chain from the probability distribution. Sampling algorithm like Gibbs sampling and Metropolis sampling are based on this approach [23, 18, 12]. This simulates a chain of examples from the distribution $Z^{-1}(\beta)\exp(-\beta E)$. The system is said to have reached equilibrium after a period of initial iterations. Hereafter, the probability of finding the system in a given state is equal to the desired probability function, and we may derive appropriate statistical moments. Samples are not uncorrelated, however, so the simulation needs to be run for long enough (which may be significant, computationally) in order to obtain a small enough variance. We shall return to the Markov Chain sampling methods later in relation to Bayesian analysis in Section 2.5.1. For now we will describe an approximation to simulated annealing, mean field annealing, which has proven to approximate the results of the simulated annealing algorithm but with a significant decrease of computation. In mean field theory, the state variables in the energy function is substituted for the mean state variables $\langle u_i \rangle$ conditioned on the remaining field mean variables at a given temperature of the system is computed iteratively. We will illustrate this with an example.

> **Example 2.3** *We will use the Markov Random Field together with mean field annealing to perform edge-preserving filtering of noisy three-dimensional MRI volumes of the human brain. The computations follow the line of Zerubia et al. [24], who invoke mean field annealing in a compound Gaussian and binary random fields estimation problem. The binary edge model is equivalent to the line-site model introduced by Geeman et al. [18], although an improved edge connectivity prior model has been added, due to Philipsen et al. [25].*

The prior distribution reflects our general insight into the objects being imaged, in this example expressing that the image represents spatially extended anatomical structures bounded by relatively sharp edges. The Weak Membrane Model, also known as the line-site model, is a simple model for reconstruction of piece-wise continuous intensity surfaces from noisy observations and was first introduced by [18]. In this model an auxiliary set of binary variables, the so-called line sites, is introduced representing discontinuities in the image. The prior formalizes the expectation that neighbor intensity values should be correlated, except when they are disconnected by the rare occasion of an active edge unit.

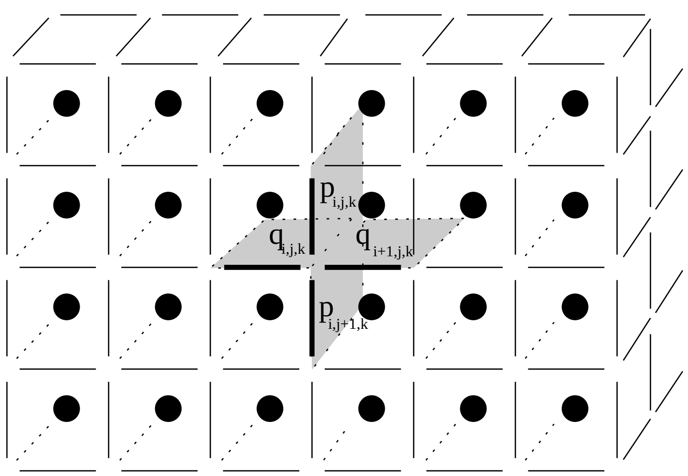


Figure 2.5: Edge elements in the Weak Membrane Model. The four shaded edge elements make up a cross; there should be an even number of active elements in a cross for proper edge geometry, as obtained by the energy term $p_{ijk}p_{i,j+1,k}q_{ijk}q_{i+1,j,k}$, which is part of the prior over the edge variables.

Extending the line-site model to three dimensions, the line-sites become edge elements p_{ijk}, q_{ijk} and r_{ijk} representing small planar elements in the i-, j- and k directions respectively, so that for example $p_{ijk} = +1$ indicates an active edge between voxels u_{ijk} and $u_{i+1,j,k}$. Inactive edge elements take the value -1. Using Gibbs distributions, the prior energy function reads

$$\begin{aligned}
&E_{prior}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}) = \\
&\quad \frac{\nu}{2}\sum_{ijk}\Big[(1-p_{ijk})(u_{ijk}-u_{i+1,j,k})^2 \\
&\quad +(1-q_{ijk})(u_{ijk}-u_{i,j+1,k})^2+(1-r_{ijk})(u_{ijk}-u_{i,j,k+1})^2\Big] \\
&\quad +\sum_{ijk}\mu^p p_{ijk}+\sum_{ijk}\mu^q q_{ijk}+\sum_{ijk}\mu^r r_{ijk} \\
&\quad -\gamma\Big[\sum_{ijk}p_{ijk}p_{i,j+1,k}q_{ijk}q_{i+1,j,k}+\sum_{ijk}p_{ijk}p_{i,j,k+1}r_{ijk}r_{i+1,j,k} \\
&\qquad +\sum_{ijk}q_{ijk}q_{i,j,k+1}r_{ijk}r_{i,j+1,k}\Big].
\end{aligned} \tag{2.55}$$

Note that the $\mu^{p..r}$-terms in the cost function control of the number of active edge-units. The γ-terms are introduced to make edges form spatially coherent surfaces as can be verified by calculating the contribution of this term for all combinations of active/inactive edge elements. For proper edge geometry there should be an even number of active elements in a cross of edge elements, as indicated in Figure 2.5. This is obtained by the energy term $p_{ijk}p_{i,j+1,k}q_{ijk}q_{i+1,j,k}$. And since we can reorient this cross in three positions, we get obtain the additional terms $p_{ijk}p_{i,j,k+1}r_{ijk}r_{i+1,j,k}$ and $q_{ijk}q_{i,j,k+1}r_{ijk}r_{i,j+1,k}$.

The degradation process produces the measurements $\boldsymbol{d}$ from $\boldsymbol{u}$. For simplicity we will study the *addition of zero-mean, white Gaussian noise with variance* σ^2

$$P(\boldsymbol{d}\,|\,\boldsymbol{u}) = Z_2^{-1}\exp\Big(-\frac{1}{2\sigma^2}\sum_{ijk}(u_{ijk}-d_{ijk})^2\Big). \tag{2.56}$$

Using Bayes formula we obtain the posterior energy function

$$E_{post}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r},\boldsymbol{d}) = \frac{1}{2\sigma^2}\sum_{ijk}(u_{ijk}-d_{ijk})^2 + E_{prior}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}). \tag{2.57}$$

Inspecting the posterior distribution we note that the three-component process $(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r})$ *realizes a Compound Random Markov Field: The posterior distribution of a single voxel* u_{ijk} *is a function of only the neighboring u-values and* d_{ijk} *This property ensures that the implementation only involves neighbor connectivity. To enhance the sampling efficiency we use a mean field annealing strategy. The scheme is implemented by introducing the temperature* $T = 1/\beta$ *in the Gibbs distribution and designing the sampler with a decreasing sequence of temperatures ending at, e.g.,* $T = 1$*. The temperature dependent distribution reads*

$$P(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}\,|\,\boldsymbol{d}) = Z^{-1}(\beta)\exp\big(-\beta E_{post}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r},\boldsymbol{d})\big) \tag{2.58}$$

As investigated by Geeman [18], sampling of this distribution is too slow for most applications. It is therefore recommended to invoke a deterministic approximation scheme, the so called mean field annealing [26, 24], which proceeds by gradually lowering the temperature T *(using linearly ascending* β*) and at each temperature computing the expectations*

$$\langle u_{ijk}, p_{ijk}, q_{ijk}, r_{ijk} \mid \mathcal{N}_{ijk}\rangle \tag{2.59}$$

$$\textit{with} \qquad \mathcal{N}_{ijk} = \big\{\langle u_{i'j'k'}|\mathcal{N}_{i'j'k'}\rangle, \langle p_{i'j'k'}|\mathcal{N}_{i'j'k'}\rangle, \tag{2.60}$$

$$\langle q_{i'j'k'}|\mathcal{N}_{i'j'k'}\rangle, \langle r_{i'j'k'}|\mathcal{N}_{i'j'k'}\rangle \mid (i'j'k') \in \mathcal{C}_{ijk}\big\} \tag{2.61}$$

iteratively. Note the mean field approximation of

$$\begin{aligned}&\big\langle u_{ijk}, p_{ijk}, q_{ijk}, r_{ijk}|u_{i'j'k'}, p_{i'j'k'}, q_{i'j'k'}, r_{i'j'k'}, \quad (i',j',k') \in \mathcal{C}_{ijk}\big\rangle\\ &\quad \simeq \big\langle u_{ijk}, p_{ijk}, q_{ijk}, r_{ijk}|\langle u_{i'j'k'}\rangle, \langle p_{i'j'k'}\rangle, \langle q_{i'j'k'}\rangle, \langle r_{i'j'k'}\rangle, \quad (i',j',k') \in \mathcal{C}_{ijk}\big\rangle.\end{aligned}$$

Since all variables are mean values, we write will use the same symbols for the averaged quantities (u^k_{ijk} *is* $\langle u_{ijk}|\mathcal{N}_{ijk}\rangle$ *at iteration* k*) as for the stochastic ones; this should not lead to confusion since the two never occur in the same expression.*

The mean field equations are thus computed as

$$\begin{aligned}u^{(k+1)}_{ijk} &= \int_u \frac{\exp\big[-E_{post}(u_{ijk}=u)\big]}{\int_{u'}\exp\big[-E_{post}(u_{ijk}=u')\big]\,du'}u\,du\\ p^{(k+1)}_{ijk} &= \frac{\exp\big[-E_{post}(p_{ijk}=+1)\big]\cdot(+1)+\exp\big[-E_{post}(p_{ijk}=-1)\big]\cdot(-1)}{\exp\big[-E_{post}(p_{ijk}=+1)\big]+\exp\big[-E_{post}(p_{ijk}=-1)\big]},\end{aligned}$$

and likewise for the $\boldsymbol{q}$ *and* $\boldsymbol{r}$ *fields. This leads to the self-consistent mean field equations for the weak membrane model (see [10] for further introduction)*

$$\begin{aligned}u^{(k+1)}_{ijk} &= \kappa^{-1}\big[\omega\, d_{ijk} + (1-p^k_{ijk})u^k_{i+1,j,k} + (1-p^k_{i-1,j,k})u^k_{i-1,j,k} + (1-q^k_{ijk})u^k_{i,j+1,k}\\ &\qquad + (1-q^k_{i,j-1,k})u^k_{i,j-1,k} + (1-r^k_{ijk})u^k_{i,j,k+1} + (1-r^k_{i,j,k-1})u^k_{i,j,k-1}\big]\\ p^{(k+1)}_{ijk} &= \tanh\Big[\beta\Big(\frac{\nu}{2}(u^{(k)}_{ijk}-u^{(k)}_{i+1,j,k})^2 - \mu^p + \gamma\,\Gamma^p(k)\Big)\Big]\\ q^{(k+1)}_{ijk} &= \tanh\Big[\beta\Big(\frac{\nu}{2}(u^{(k)}_{ijk}-u^{(k)}_{i,j+1,k})^2 - \mu^q + \gamma\,\Gamma^q(k)\Big)\Big]\\ r^{(k+1)}_{ijk} &= \tanh\Big[\beta\Big(\frac{\nu}{2}(u^{(k)}_{ijk}-u^{(k)}_{i,j,k+1})^2 - \mu^r + \gamma\,\Gamma^r(k)\Big)\Big]\end{aligned} \tag{2.62}$$

where $\beta = 1/T$, $\omega = \sigma^{-2}$ *and*

$$\begin{aligned}
\Gamma^p_{ijk}(k) &= p^k_{i,j+1,k} q^k_{ijk} q^k_{i+1,j,k} + r^k_{ijk} r^k_{i+1,j,k} p^k_{i,j,k+1} && (2.63)\\
\Gamma^q_{ijk}(k) &= p^k_{ijk} p^k_{i,j+1,k} q^k_{i+1,j,k} + r^k_{ijk} r^k_{i,j+1,k} q^k_{i,j,k+1} && (2.64)\\
\Gamma^r_{ijk}(k) &= p^k_{ijk} p^k_{i,j,k+1} r^k_{i+1,j,k} + q^k_{ijk} q^k_{i,j,k+1} r^k_{i,j+1,k} && (2.65)\\
\kappa &= \omega + 6 - p^k_{ijk} - p^k_{i-1,j,k} - q^k_{ijk} - q^k_{i,j-1,k} - r^k_{ijk} - r^k_{i,j,k-1}. && (2.66)
\end{aligned}$$

The coupled equations are solved by straightforward iteration as shown in Equation (2.62), defining a recursive nearest neighbor connected cellular neural network.

A direct implementation of the updating scheme for the MRF presented in the previous section is memory expensive because an update of the image volume and the edge volumes requires both the old iteration and a new copy. Because the updating schedule only depends on the nearest neighbors it is possible to update only half of the elements in each iteration in an alternating checkerboard fashion, see Figure 2.6. This updating scheme makes it possible to use the same array for the k'*th and* $k+1'$*th iteration thereby saving memory. Furthermore, the speed of convergence of the Markov field is doubled, since the part of the field that is being updated experiences the field of the next iteration. A t1-weighted MRI volume (MPRAGE acquisition sequence) consisting of*

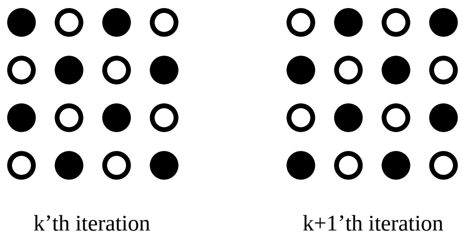


Figure 2.6: Updated elements are marked with black.

$160 \times 180 \times 180$ *voxels of size* $1 \times 1 \times 1\, mm^3$ *is used as the input image for the Markov Random Field. The goal is to attain coherent edge surfaces in the image, in order to identify anatomical regions. The t1-weighted image is chosen because it gives good gray/white-matter contrast which is of special interest in functional brain imaging. The parameters used for the experiment were noise parameter* $\omega = 0.5$, *edge coherency* $\gamma = 7.4$, *edge penalties* $\mu_{p,q,r} = 15.0$, *object smoothness* $\nu = 2$, *and the annealing scheme was 100 iterations with* β *ranging linearly from* 0.001 *to* 0.06. *Slices of the resulting edge images* $\boldsymbol{p}, \boldsymbol{q}$ *and* $\boldsymbol{r}$ *are shown in Figure 2.7 together with the original and noise reduced MR image slices. The noise reduced image may prove to be more suitable for segmentation than the original image because of the reduction of noise. The edges are seen to be coherent in a large scale, and illustrates the effectiveness of the Markov Field as an image processing tool.*

■

Example 2.4 PET smoothing with MRI Attained Edge Map. *The mean field equations of the previous example can be run with 'fixed' edge elements. This is relevant for noise filtering in PET as will be demonstrated in this example. In functional activation studies it can be desirable to reduce the resolution in the image in order to decrease the noise level. This is typically performed by filtering the image volume with a low-pass filtering kernel, i.e. a Gaussian kernel or a boxcar filter. Spatial smoothing also allows for inter-subject functional differences to be less pronounced, since the effect of registration errors is reduced with decreasing resolution. The low-pass filtering does however cause signals originating from regions that may be anatomically separated by one or more tissue boundaries (cortical fissures) to be mixed in the analysis, possibly concealing potentially activated areas. Furthermore, the filtering of activity into nearby regions with very low activity (e.g. CSF) tends to bias the activation signals near such anatomical boundaries towards zero. To remedy this situation, we can use the MRF edge filtering model using edges obtained from an MRI scan.*

■

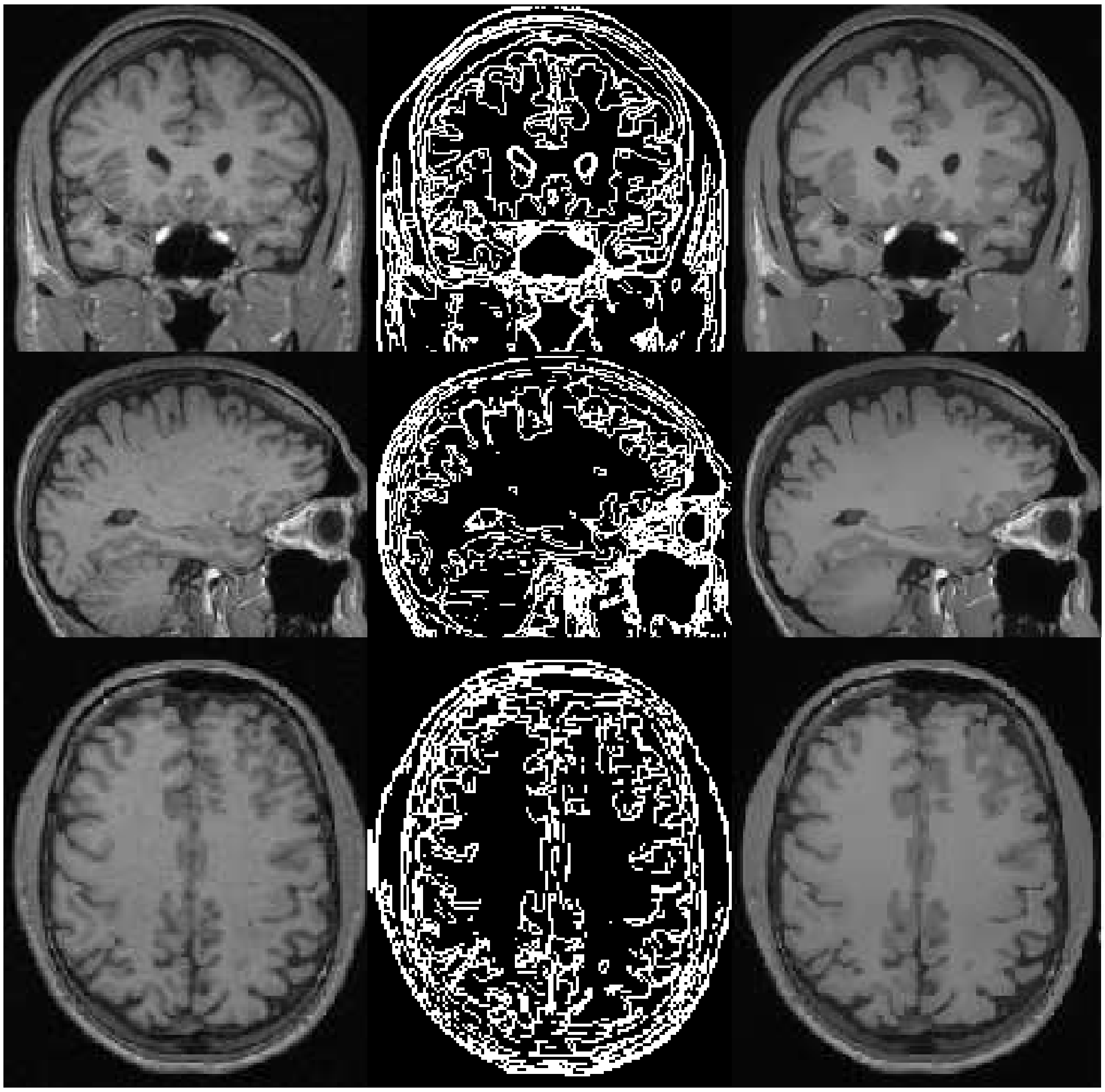


Figure 2.7: The three columns show the raw MRI volume, resulting edge map and noise reduced volume of Example 3.

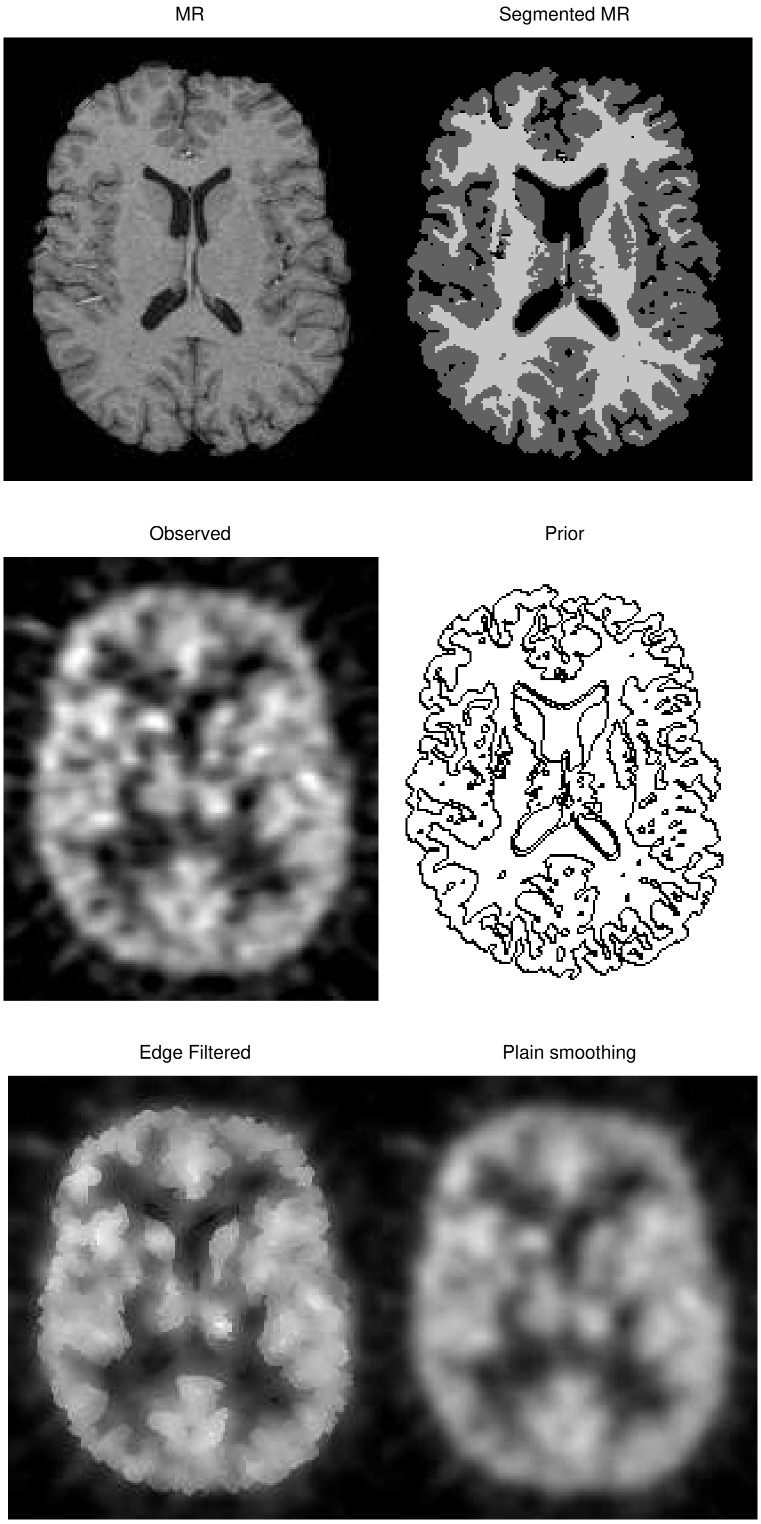


Figure 2.8: Top row: Stripped t1 and segmented MR slice, second row: observed PET image slice and edge prior map, last row: the resulting fixed edge-MRF smoothed image and and image with no active edges in the MRF.

2.5 Parameter Estimation In Gibbs Distributions

2.5.1 Bayesian Analysis

The fact that our model parameters are determined as a function of observed data which are a stochastic in nature makes the model parameters stochastic variables themselves. The posterior distribution $P(\theta \mid \boldsymbol{d})$ expresses this. Deciding upon a single set of parameter values $\boldsymbol{\theta}_{\mathrm{ML}}$, for example like the maximum likelihood estimate, essentially throws away precious information from our training examples. In Bayesian analysis the philosophy is that we should never truncate distributions in this manner. Rather, all variables are associated with their respective probability distributions based on accumulated knowledge. Bayesian methods for data fitting have been investigated by MacKay [27, 28] focusing on analytical approximations applied to Bayesian analysis, and by Neal [12] who investigates a stochastic approach using Markov Chain sampling.

While true Bayesian analysis may be difficult to implement for practical applications, the Bayesian analysis offers a *framework* in which classical statistical methods can be understood and regarded as approximations to the Bayesian ideal. We will in the following mainly use the stochastic sampling approach to the Bayesian analysis where possible. It will be seen how Maximum Likelihood (ML) estimation can be seen as a special approximation to the Bayesian integrals, and especially how a Bayesian viewpoint can argue for the use of log-transformations for ML-estimation of certain model parameters; the so called scale parameters.

Now, consider the case where we infer the parameters for an image restoration problem using the same image $\boldsymbol{d}$ that we wish to restore (alternatively, one may infer the model parameters on basis of a set of similar image examples $\{\boldsymbol{d}^{1..P}\}$). The posterior distribution for the model parameters reads $P(\boldsymbol{\theta} \mid \boldsymbol{d})$; this uncertainty in what parameter values to use for the actual image restoration procedure increases the uncertainty of the underlying 'true' image $\boldsymbol{u}$ given the observed image $\boldsymbol{d}$. This can expressed formally by averaging over all possible parameter values

$$P(\boldsymbol{u} \mid \boldsymbol{d}) = \int P(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{d})\, d\boldsymbol{\theta} = \int P(\boldsymbol{u} \mid \boldsymbol{d}, \boldsymbol{\theta}) P(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{2.67}$$

and by further applying Bayes relation (2.44) we obtain

$$P(\boldsymbol{u} \mid \boldsymbol{d}) = P(\boldsymbol{d})^{-1} \int P(\boldsymbol{d} \mid \boldsymbol{u}, \boldsymbol{\theta}) P(\boldsymbol{u} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})\, d\boldsymbol{\theta}. \tag{2.68}$$

We readily identify the integrand as the product of the degeneration model, the object prior model and the parameter prior model. Inserting the appropriate energy functions we have

$$P(\boldsymbol{u} \mid \boldsymbol{d}) = \frac{\int \exp\left[-E_{degen}(\boldsymbol{d}, \boldsymbol{u}, \boldsymbol{\theta}) - E_{prior,obj}(\boldsymbol{u}, \boldsymbol{\theta}) - E_{prior,\theta}(\boldsymbol{\theta})\right] d\boldsymbol{\theta}}{P(d) Z_{degen}(\boldsymbol{\theta}) Z_{prior,obj}(\boldsymbol{\theta}) Z_{prior,\theta}(\boldsymbol{\theta})} \tag{2.69}$$

The term $P(\boldsymbol{d})$ which can be expressed as $P(\boldsymbol{d}) = \int P(\boldsymbol{u} \mid \boldsymbol{d}, \boldsymbol{\theta}) P(\boldsymbol{u} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})\, d\boldsymbol{u}\, d\boldsymbol{\theta}$ is considered undetermined in the following. This term, known as the evidence, is relevant when discussing model comparison, see section 2.5.2. But first we shall take a look at methods for evaluating the posterior distribution $P(\boldsymbol{u} \mid \boldsymbol{d})$.

The Markov Chain Monte Carlo method

An unbiased estimate of the Bayesian Posterior distribution can be obtained using Markov Chain Monte Carlo simulation (MCMC) of the posterior distribution. This method is powerful although computationally quite demanding. The integration is performed by performing a 'random walk' (similar to the simulated annealing discussed earlier) in all variables and parameters, e.g. for the image restoration problem $\boldsymbol{u}, \boldsymbol{\theta}$, according to the simultaneous distribution $P(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{d}) = P(\boldsymbol{u} \mid \boldsymbol{d}, \boldsymbol{\theta}) P(\boldsymbol{\theta})$ thus generating samples from the distribution. With such samples in hand one may easily deduce the most probable restored image $\boldsymbol{u}$ and the error bars associated with it. The random walk is obtained by transforming the probability function we wish to simulate into a Markov Chain: Let the state of the Markov chain define a sample of the distribution. The next state of the Markov chain is determined by perturbing the current state according to the desired distribution, defining the transition probabilities of the Markov Chain.

A simple implementation Markov Chain is the Gibbs sampler. Each element of the state variable is updated in turn, and the transition probability is identical to the conditional distribution of the state element given the remaining state elements. For variables that take discrete values, the simulation can be performed by simply evaluating the conditional probability of each discrete value and randomly updating the state element accordingly. For continuously valued variables the updating may be performed if the conditional distribution is of a standard form, e.g. Gaussian. If the conditional distribution is not of a standard form the Gibbs sampler can be difficult to implement.

The Metropolis algorithm for Markov Chain sampling [23] avoids any need to sample from difficult distributions. If can be applied to any problem, continuous or discrete, as long as the ratio of the probability of two states can be computed. The algorithm works by repeatedly generating a candidate step $(\boldsymbol{u}_+, \boldsymbol{\theta}_+)$ by randomly perturbing $(\boldsymbol{u}, \boldsymbol{\theta})$. If $P(\boldsymbol{u}_+, \boldsymbol{\theta}_+ \,|\, \boldsymbol{d}) > P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d})$, the candidate step is accepted, otherwise the candidate step is accepted with probability $P(\boldsymbol{u}_+, \boldsymbol{\theta}_+ \,|\, \boldsymbol{d})/P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d})$. Note that all rejected as well as accepted steps are included in the collection of samples. The advantage of this algorithm in relation to Gibbs distributions is that the value of the partition function does not need to be computed, since the partition function is canceled in the ratio of state probabilities: $P(\boldsymbol{u}_+, \boldsymbol{\theta}_+ \,|\, \boldsymbol{d})/P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d}) = \exp(E_+ - E)$. This, of course, provided that the partition function does not change with the state.

In image processing the partition function usually depends on the parameter vector so the ratio of the partition function needs to be known which is the major difficulty for applying the sampling scheme for parameter estimation. Note that for some parameters, the partition function can be expressed in a closed form. An example is the often used Gaussian white noise degeneration model $P(\boldsymbol{d} \,|\, \boldsymbol{u}, \sigma) = Z(\sigma)^{-1} \exp\big[-(\boldsymbol{u} - \boldsymbol{d})^T(\boldsymbol{u} - \boldsymbol{d})/2\sigma^2\big]$ where we have $Z(\sigma) = \sqrt{(2\pi\sigma^2)^N}$.

Auxiliary variables similar to the line site elements described in Example 2.3 may be included in the vector $\boldsymbol{u}$. The sampler should be made to exploit the Markov property of the image models (i.e. the conditional distribution of a single voxel is only a function of voxels in a local clique) thus making the algorithm computationally feasible. A complication arises when the partition function $Z(\boldsymbol{\theta})$ depends on the parameter vector $\boldsymbol{\theta}$ in a way that cannot be expressed in closed form. A solution to this is to exclude the problematic parameters from the random walk, if these parameters can be determined by other methods. This is not quite in line with the Bayesian philosophy, a more correct approach is to estimate the partition function's dependency of the parameter as performed by Higdon et al. [29, 17].

To implement the Metropolis algorithm we need to choose a distribution of candidate states, the so called proposal distribution $S(\boldsymbol{x}_+, \boldsymbol{x})$, $\boldsymbol{x}$ being the collection of state variables, which allows some tailoring to the problem to be introduced. We may, for example, update the state elements in turn, in random order, or all together in each step, e.g. using a multivariate Gaussian distribution. Which method is more appropriate depends on the distribution: When updating elements one at a time, the algorithm will tend to have difficulties penetrating barriers (areas in state space with a low probability that separate areas with high probability, e.g. multi modal distributions) that are on the diagonal in state space, i.e. jumping from one basin to another requires the collective change of a number of state variables. Probability density functions of such nature arise when the energy functions are complex (non-linear) functions of the state. On the other hand, if the state variables are largely uncorrelated, the individual updates give near optimal performance. Another aspect of the proposal distribution is its variance. If the variance is too small, the sampler will has a tendency to get stuck in the same local minimum of the energy function and the individual samples produced will be highly correlated so that a large number of samples is required to fully explore the mode. On the other hand, if the variance is too high, most of the candidates will be rejected so that the computing time will be spent generating rejected candidates. In any case, the proposal distribution will have to be determined using either a priori knowledge of the problem or by trial and error. See [12] for further discussion of different candidate distribution.

Mean Field versus. Metropolis

In Example 2.3 it was demonstrated how the mean field algorithm combined with an annealing schedule was able to find usable edge maps, essentially solving a large combinatorial problem. In addition, the algorithm found a solution for 4 times $160 \times 180 \times 180 \simeq 20$ million variables in only 100 global iterations. Application of the Metropolis sampling scheme would require many more iterations in order for the system to reach equilibrium and obtain enough samples from the steady state to determine the mode. It is evident that the mean field approach is applicable to a problem only when the individual conditional distributions of state elements given the neighboring state elements is unimodal. The mean field algorithm does its work at the expense of throwing away information regarding the uncertainty of the solution obtained. Therefore it can only be used to approximate the location of the global minimum of the energy function, its solution does not tell us anything regarding the width of the surrounding basin, so the mean field algorithm cannot be used for implementing a true Bayesian analysis.

2.5.2 Bayesian Model Comparison

It is possible to take the Bayesian analysis one step further and perform what is known as Bayesian model comparison, which—at least in theory—allows us to select among different models and model architectures by evaluating the posterior probability of each model. A model in this context is understood as the collection of model architecture (e.g. size of region of support for an AR model), error function, noise model etc., in other words, everything but the value of the model parameters. We designate the model by the symbol $\mathcal{M}_i$, subscripted to indicate that the model is one of a set of candidate models $\{\mathcal{M}_{1..M}\}$. We wish to compare each model's posterior probability given the observed data

$$P(\mathcal{M}_i \,|\, \boldsymbol{d}) = \frac{P(\boldsymbol{d} \,|\, \mathcal{M}_i) P(\mathcal{M}_i)}{P(\boldsymbol{d})} \tag{2.70}$$

If we have no preference of models $P(\mathcal{M}_i)$ we evaluate the *model evidence* $P(\boldsymbol{d} \,|\, \mathcal{M}_i)$

$$P(\boldsymbol{d} \,|\, \mathcal{M}_i) \quad = \quad \int P(\boldsymbol{d}, \boldsymbol{u}, \boldsymbol{\theta} \,|\, \mathcal{M}_i) \, d\boldsymbol{u} \, d\boldsymbol{\theta} = \int P(\boldsymbol{d} \,|\, \boldsymbol{u}, \boldsymbol{\theta}, \mathcal{M}_i) P(\boldsymbol{u} \,|\, \boldsymbol{\theta}, \mathcal{M}_i) P(\boldsymbol{\theta} \,|\, \mathcal{M}_i) \, d\boldsymbol{u} \, d\boldsymbol{\theta} \tag{2.71}$$

where we identify the object-, parameter prior- and degeneration models. Note that the model evidence corresponds to the normalization constant of (2.68). An example of Bayesian model comparison in the context of feed-forward neural networks is given in [27, 28], where the posterior distribution is approximated by quadratic functions. The application of the model comparison framework to image restoration provides a consistent scheme for selection among different model architectures in the absence of 'ground truth' examples $\boldsymbol{u}$. But the evaluation of the above integral is difficult in general without the introduction of substantial approximations. We shall not consider this issue further, although the subject is well worth further investigation.

2.5.3 Maximum Likelihood Estimation

Depending on the problem at hand we can apply appropriate approximations to the posterior distribution of the observed image, i.e. integral (2.68). When the degrees of freedom in the observed data[4] is far larger than the number of model parameters, the distribution $P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d})$ is likely to be highly peaked with respect to the parameter vector $\boldsymbol{\theta}$ with the peak localized at $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{ML}}$. This expresses the fact that the parameters are well determined from the available data. We may thus replace $P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d}) = P(\boldsymbol{\theta} \,|\, \boldsymbol{d}) P(\boldsymbol{u} \,|\, \boldsymbol{\theta}, \boldsymbol{d})$ by $P(\boldsymbol{u} \,|\, \boldsymbol{d}, \boldsymbol{\theta}) \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{ML}})$ so that the posterior distribution (2.68) becomes

$$P(\boldsymbol{u} \,|\, \boldsymbol{d}) = \int P(\boldsymbol{u}, \boldsymbol{\theta} \,|\, \boldsymbol{d}) \, d\boldsymbol{\theta} \simeq \int P(\boldsymbol{u} \,|\, \boldsymbol{d}, \boldsymbol{\theta}) \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{ML}}) \, d\boldsymbol{\theta} = P(\boldsymbol{u} \,|\, \boldsymbol{d}, \boldsymbol{\theta}_{\text{ML}}) \tag{2.72}$$

Determining the value $\boldsymbol{\theta}_{\text{ML}}$ that maximizes $P(\boldsymbol{\theta} \,|\, \boldsymbol{d})$ is equal to maximizing the product $P(\boldsymbol{d} \,|\, \boldsymbol{\theta}) P(\boldsymbol{\theta})$, where $P(\theta)$ is the prior over parameter values and is typically either assumed flat or Gaussian. Including such a prior term yields the so called penalized likelihood $L(\boldsymbol{\theta}; \boldsymbol{d}) = P(\boldsymbol{d} \,|\, \boldsymbol{\theta}) P(\boldsymbol{\theta})$ which reads

$$\begin{aligned} L(\boldsymbol{\theta}; \boldsymbol{d}) = P(\boldsymbol{d} \,|\, \boldsymbol{\theta}) P(\boldsymbol{\theta}) &= \int P(\boldsymbol{d}, \boldsymbol{u} \,|\, \boldsymbol{\theta}) P(\boldsymbol{\theta}) \, d\boldsymbol{u} \\ &= \int P(\boldsymbol{d} \,|\, \boldsymbol{u}, \boldsymbol{\theta}) P(\boldsymbol{u} \,|\, \boldsymbol{\theta}) P(\boldsymbol{\theta}) \, d\boldsymbol{u} = Z(\boldsymbol{\theta})^{-1} \int \exp\big[- E_{degen}(\boldsymbol{d}, \boldsymbol{u}, \boldsymbol{\theta}) - E_{prior}(\boldsymbol{u}, \boldsymbol{\theta}) - E_{prior,\theta}(\boldsymbol{\theta}) \big] \, d\boldsymbol{u} \end{aligned}$$

where $Z(\boldsymbol{\theta}) = Z_{degen}(\boldsymbol{\theta}) Z_{prior}(\boldsymbol{\theta}) Z_{prior,\theta}(\boldsymbol{\theta})$. A flat prior is implemented by simply excluding $P(\boldsymbol{\theta})$ (and $Z_{prior,\theta}(\boldsymbol{\theta})$). The classical approach of minimizing $-\log L(\boldsymbol{\theta}; \boldsymbol{d})$ directly is apparently a more complex problem for the Gibbs class of probability distributions.

In this section we shall present a general learning rule for the training of parameters in Gibbs distributions. The so called Boltzmann Learning rule can be used for training parameters by maximizing the above likelihood.

[4] Note that, since we typically expect the observed data to contain a large degree of auto-correlation, we cannot simply count the total number of pixels is observed example images.

The Generalized Boltzmann Learning Rule

Consider a dynamical system which is partially observed, meaning that only part of the systems state variables can be measured. Denoting the state of the system $\boldsymbol{x} = \{\boldsymbol{u}, \boldsymbol{d}\}$ the partitioning of the state space into visible and hidden part is $\boldsymbol{d}$ and $\boldsymbol{u}$, respectively. In relation to the image restoration problem, the observed state represents the observed degraded image, and the unobserved state variable is the underlying object and possibly the auxiliary variables similar to the edge elements of example 2.3. Our model of the state probability distribution is assumed to take Gibbs form

$$P_\theta(\boldsymbol{x}) = Z(\boldsymbol{\theta})^{-1} \exp\big[-E(\boldsymbol{x};\boldsymbol{\theta})\big]. \tag{2.73}$$

The objective is to make the model distribution $P_\theta(\boldsymbol{d})$ parameterized by $\boldsymbol{\theta}$, which we will refer to as the *student distribution*, resemble the unknown *teacher distribution* $P_*(\boldsymbol{d})$. We do not treat the parameter vector $\boldsymbol{\theta}$ as a random variable—hence we use the notation $P_\theta(\boldsymbol{x})$ rather than $P(\boldsymbol{x}\,|\,\boldsymbol{d})$.

Since we by definition have no samples of the unobserved state vector available, we need to assume that the teacher distribution of this part of the state vector $P_*(\boldsymbol{u})$ is identical to the student distribution $P_\theta(\boldsymbol{u})$. We define $P_* = P_*(\boldsymbol{x}) = P_*(\boldsymbol{d})P_\theta(\boldsymbol{u}\,|\,\boldsymbol{d})$, just like $P_\theta = P_\theta(\boldsymbol{x}) = P_\theta(\boldsymbol{d})P_\theta(\boldsymbol{u}\,|\,\boldsymbol{d})$.

The Kullbak information distance is a measure of the difference between two probability distributions, and is defined by

$$\Delta[P_*, P_\theta] = \left\langle \log \frac{P_*}{P_\theta} \right\rangle_* = \int P_* \log \frac{P_*}{P_\theta}\, d\boldsymbol{x} \tag{2.74}$$

The notation $\langle \cdot \rangle_*$ and $\langle \cdot \rangle_\theta$ indicates the mean of $\cdot$ with respect to the distributions $P_*(\boldsymbol{x})$ and $P_\theta(\boldsymbol{x})$, respectively. The Kullbak distance $\Delta[P_*, P_\theta]$ is positive or zero, and assumes the zero value only when the two distributions are identical. The measure is positive as can be seen from $\log x > 1 - 1/x$, so that $\Delta \geq \int (P_* - P_\theta)\, d\boldsymbol{x} = 0$.

The Boltzmann Learning rule expresses that the Kullbak distance can be minimized by updating student parameters according to the gradient

$$\boxed{\frac{\partial \Delta[P_*, P_\theta]}{\partial \boldsymbol{\theta}} = \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_* - \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_\theta} \tag{2.75}$$

The second derivative may be computed using

$$\boxed{\begin{aligned} \frac{\partial^2 \Delta[P_*, P_\theta]}{\partial \theta_i \partial \theta_j} \;=\; & \left\langle \frac{\partial^2 E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right\rangle_* + \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i} \cdot \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j} \right\rangle_\theta \\ & - \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i} \right\rangle_\theta \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_j} \right\rangle_\theta - \left\langle \frac{\partial^2 E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right\rangle_\theta \end{aligned}} \tag{2.76}$$

The above equation originates from the Boltzmann Machine Learning rule for parameter learning in Boltzmann Machines and was first introduced by Hinton and Sejnowski [13, 14] and has led to many applications of the Boltzmann Machine. A more general formulation like the one presented here was introduced in [15]. *Proof of learning rule:*

$$\frac{\partial \Delta[P_*, P_\theta]}{\partial \boldsymbol{\theta}} = -\int P_* \frac{1}{P_\theta} \frac{\partial P_\theta}{\partial \boldsymbol{\theta}}\, d\boldsymbol{x} \tag{2.77}$$

Now, using $P_\theta = Z^{-1}\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))$

$$\begin{aligned} \frac{\partial P_\theta}{\partial \boldsymbol{\theta}} &= -\frac{\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))}{Z}\Big[\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{1}{Z}\frac{\partial Z}{\partial \boldsymbol{\theta}}\Big] \\ &= -P_\theta\Big[\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{1}{Z}\frac{\partial Z}{\partial \boldsymbol{\theta}}\Big] \end{aligned} \tag{2.78}$$

To find $\partial Z/\partial\boldsymbol{\theta}$ we recall that $Z = \int \exp(-E(\boldsymbol{x};\boldsymbol{\theta}))\,d\boldsymbol{x}$ so that

$$\begin{aligned}\frac{\partial Z}{\partial\boldsymbol{\theta}} &= \int -\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\exp(-E(\boldsymbol{x};\boldsymbol{\theta}))\,d\boldsymbol{x}\\ &= -Z\Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_\theta\end{aligned}$$

Inserting this into (2.78) yields

$$\frac{\partial P_\theta}{\partial\boldsymbol{\theta}} = -P_\theta\left[\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_\theta\right] \tag{2.79}$$

and inserting this into (2.77) we obtain

$$\begin{aligned}\frac{\partial\Delta[P_*,P_\theta]}{\partial\boldsymbol{\theta}} &= \int P_*\left[\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_\theta\right]d\boldsymbol{x} &(2.80)\\ &= \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_* - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_\theta &(2.81)\end{aligned}$$

which proves (2.75). The second derivative is found using (2.80)

$$\frac{\partial^2\Delta[P_*,P_\theta]}{\partial\theta_i\partial\theta_j} = \frac{\partial}{\partial\theta_j}\int P_*\left[\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i} - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}\Big\rangle_\theta\right]d\boldsymbol{x} \tag{2.82}$$

and since

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}\Big\rangle_\theta &= \int\frac{\partial}{\partial\theta_j}\left(P_\theta\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}\right)d\boldsymbol{x} = \int\left[\frac{\partial P_\theta}{\partial\theta_j}\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i} + P_\theta\frac{\partial^2E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\right]d\boldsymbol{x} &(2.83)\\ &= \int\left[P_\theta\frac{\partial^2E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j} - \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}P_\theta\left(\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_j} - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_j}\Big\rangle_\theta\right)\right]d\boldsymbol{x} &(2.84)\\ &= \Big\langle\frac{\partial^2E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}\Big\rangle_\theta - \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_j}\Big\rangle_\theta + \Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_i}\Big\rangle_\theta\Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\theta_j}\Big\rangle_\theta &(2.85)\end{aligned}$$

wee see that (2.76) follows.

■

Remarks:

1. Minimization of the Kullbak distance $\Delta[P_*(\boldsymbol{x}),P_\theta(\boldsymbol{x})]$, measured on the full state vector $\boldsymbol{x} = (\boldsymbol{u},\boldsymbol{d})$ with a partially observed teacher distribution $P_*(\boldsymbol{x}) = P_*(\boldsymbol{d})P_\theta(\boldsymbol{u}\,|\,\boldsymbol{d})$ is identical to the minimization of the Kullbak distance measured only on the observed state $\Delta[P_*(\boldsymbol{d}),P_\theta(\boldsymbol{d})]$:

$$\begin{aligned}\Delta[P_*(\boldsymbol{x}),P_\theta(\boldsymbol{x})] &= \int P_*(\boldsymbol{x})\log\frac{P_*(\boldsymbol{x})}{P_\theta(\boldsymbol{x})}\,d\boldsymbol{x} = \int P_*(\boldsymbol{d})P_\theta(\boldsymbol{u}\,|\,\boldsymbol{d})\log\frac{P_*(\boldsymbol{d})}{P_\theta(\boldsymbol{d})}\,d\boldsymbol{u}\,d\boldsymbol{d}\\ &= \int P_*(\boldsymbol{d})\log\frac{P_*(\boldsymbol{d})}{P_\theta(\boldsymbol{d})}\,d\boldsymbol{d} = \Delta[P_*(\boldsymbol{d}),P_\theta(\boldsymbol{d})] &(2.86)\end{aligned}$$

2. The teacher distribution of the observed variable $P_*(\boldsymbol{d})$ can be expressed from a finite set of examples by representing each example with a delta function $\boldsymbol{d}^{(1..P)}$, $P_*(\boldsymbol{d}) = \frac{1}{P}\sum_{p=1}^P\delta(\boldsymbol{d}-\boldsymbol{d}^{(p)})$. The average $\langle\partial E(\boldsymbol{x};\boldsymbol{\theta})/\partial\boldsymbol{\theta}\rangle_*$ with respect to the teacher distribution should be computed by clamping the observed state variable to the observed examples $\boldsymbol{d}^{(p)}$. For each example, the hidden state vector $\boldsymbol{u}$ is sampled according to the student distribution $P(\boldsymbol{u}\,|\,\boldsymbol{d})$, either by analytical integration/summation over all state combinations or by simulation, e.g. using a sampling scheme like Gibbs sampling or Metropolis sampling.

   With the above definition of the teacher distribution, the teacher average is computed as an average over examples:

$$\Big\langle\frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_* = \frac{1}{P}\sum_{p=1}^P\Big\langle\frac{\partial E(\boldsymbol{d}^{(p)},\boldsymbol{u};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\Big\rangle_{\boldsymbol{u}\in P_\theta(\boldsymbol{u}\,|\,\boldsymbol{d}^{(p)})} \tag{2.87}$$

3. The average $\langle \partial E(\boldsymbol{x};\boldsymbol{\theta})/\partial\boldsymbol{\theta}\rangle_\theta$ with respect to the student distribution should be computed by sampling all state variables according this distribution. This may be performed analytically if possible, otherwise using a sampling scheme.

4. When using Markov Chain sampling schemes (Gibbs or Metropolis sampling) an annealing scheme should be applied. This is to ensure that the Markov Chain reaches equilibrium state. Only samples produced at equilibrium should be used for computing averages.

5. Minimizing the Kullbak distance between the teacher distribution $P_*(\boldsymbol{d})$ and the student distribution $P_\theta(\boldsymbol{d})$ is identical to maximum likelihood estimation of $\boldsymbol{\theta}$. This can be seen by direct insertion of the teacher distribution into the definition of the Kullbak distance:

$$\begin{aligned} \arg\min_{\boldsymbol{\theta}} \Delta[P_*(\boldsymbol{d}), P_\theta(\boldsymbol{d})] &= \arg\min_{\boldsymbol{\theta}} \int \frac{-1}{P}\sum_{p=1}^{P} \delta(\boldsymbol{d}-\boldsymbol{d}^{(p)}) \log P_\theta(\boldsymbol{d})\, d\boldsymbol{d} \\ &= \arg\max_{\boldsymbol{\theta}} \frac{1}{P}\sum_{p=1}^{P} \log P_\theta(\boldsymbol{d}^{(p)}) = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta};\boldsymbol{d}) \end{aligned} \tag{2.88}$$

The Kullbak derivatives (2.75, 2.76,) corresponds to the gradient of the likelihood, hence

$$\frac{\partial\Delta}{\partial\boldsymbol{\theta}} = -\frac{\partial L(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \tag{2.89}$$

6. We may introduce a prior on the model parameters $P(\theta)$. Since $\Delta(P_*, P_\theta) = \langle \log P_*(x)\rangle_* - \langle \log P_\theta(\boldsymbol{x})\rangle_*$ we see that

$$\langle \log P(\theta \,|\, x)\rangle_* = -\Delta(\theta, \theta^*) + \log P(\theta) + c, \tag{2.90}$$

i.e. the gradient receives an extra term:

$$\frac{\partial\Delta[P_*, P_\theta]}{\partial\boldsymbol{\theta}} = \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \right\rangle_* - \left\langle \frac{\partial E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \right\rangle_\theta - \frac{\partial \log P(\theta)}{\partial\theta} \tag{2.91}$$

and the second derivative the term

$$-\frac{\partial^2 P(\boldsymbol{\theta})}{\partial\theta_i\partial\theta_j}. \tag{2.92}$$

7. If the Boltzmann learning rule is used as a gradient descent algorithm to find the maximum likelihood estimate for the parameter vector as outlined above, the samples of the teacher distribution (with the observed state clamped, i.e. using $P(\boldsymbol{u} \,|\, \boldsymbol{d}^{(p)})$) can be reused to compute the statistics of the hidden state vector. In image processing, this includes error bars on the restored image.

8. The mean field approximation of Peterson et al. [30] cannot be used for simulation of an entirely 'free' running system as required for obtaining averages over the student distribution. It may be possible (though not in general) to apply the mean field strategy for approximating the necessary averages for the teacher distribution, since these only require sampling from the conditional distribution $P_\theta(\boldsymbol{u}|\boldsymbol{d}^{(p)})$. This, on the other hand, eliminates the possibility to generate other statistics of the restored image, e.g. error bars.

9. The derivatives can be used in connection with Newton's second order method [8, 10] to accelerate the learning process[5] as demonstrated by Pedersen et al.[31]

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \mathbf{H}^{-1}\frac{\partial\Delta}{\partial\boldsymbol{\theta}}, \qquad \mathbf{H} = \frac{\partial^2 E(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T} \tag{2.93}$$

[5]Please note the vector notation for derivatives defined by $\frac{\partial E}{\partial\boldsymbol{\mu}} = (\frac{\partial E}{\partial\mu_1}, ..., \frac{\partial E}{\partial\mu_1})$, $\frac{\partial E}{\partial\boldsymbol{\mu}^T} = (\frac{\partial E}{\partial\boldsymbol{\mu}})^T$, and $\frac{\partial^2 E}{\partial\boldsymbol{\mu}^2} = \frac{\partial^2 E}{\partial\boldsymbol{\mu}\partial\boldsymbol{\mu}^T} = \mathbf{H}$, where $H_{ij} = \frac{\partial^2 E}{\partial\mu_i\partial\mu_j}$

The Newton method requires that the matrix of second derivatives has certain properties for the learning algorithm to work; namely that the matrix is non-singular and that the matrix is positive definite, i.e. all eigenvalues are positive. Very small positive eigenvalues also pose problems, the eigenvalues of the inverse matrix are the inverses of the original matrix eigenvalues, with very large steps sizes as a result. This can be helped by 'regularizing' the second derivative matrix by adding a small positive constant along the diagonal before inversion.

Example 2.5 *Learning the Mean of a Multivariate Gaussian Distribution In this example we will show the derivatives for estimating the mean of a multivariate Gaussian distribution,* $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{\mu}_*, \Sigma_*)$. *We will consider the case where the covariance matrix* Σ_* *is known. We know that the posterior distribution of the mean* $\boldsymbol{\mu}$*, given a finite set of* P *examples, is itself Gaussian distributed, i.e.*

$$\boldsymbol{\mu}_{ML} = \frac{1}{P}\sum x^{(p)} \in \mathcal{N}(\boldsymbol{\mu}_*, \Sigma/P) \tag{2.94}$$

Typically in parameter estimation problems, the posterior distribution of the parameters can be well approximated with a Gaussian distribution in regions of parameter space close to a maximum. Therefore, the behavior of the derivatives $\partial\Delta/\partial\boldsymbol{\theta}$ *and* $\partial^2\Delta/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T$ *are informative for understanding the dynamics of learning in more complicated distributions.*

The first derivative of the energy function associated with the multivariate Gaussian distribution is

$$E(\boldsymbol{x};\boldsymbol{\mu}) = (\boldsymbol{\mu}-\boldsymbol{x})^T\Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{x}) \tag{2.95}$$

$$\frac{\partial E(\boldsymbol{x};\boldsymbol{\mu})}{\partial\boldsymbol{\mu}} = \Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{x}) \tag{2.96}$$

We can perform the average of this expression analytically with respect to the teacher and student distributions

$$\left\langle \frac{\partial\Delta}{\partial\boldsymbol{\mu}} \right\rangle_* = \Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_*) \tag{2.97}$$

$$\left\langle \frac{\partial\Delta}{\partial\boldsymbol{\mu}} \right\rangle_\mu = \Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}) = 0 \tag{2.98}$$

and we readily see that $\partial\Delta/\partial\boldsymbol{\mu} = \Sigma^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_*)$ *is zero only when* $\boldsymbol{\mu} = \boldsymbol{\mu}_*$.

To find the second derivative $\partial^2\Delta/\partial\boldsymbol{\mu}^2$*, we see that* $\partial^2 E/\partial\boldsymbol{\mu}\partial\boldsymbol{\mu}^T = \Sigma^{-1}$ *and hence*

$$\left\langle \frac{\partial^2 E}{\partial\boldsymbol{\mu}\partial\boldsymbol{\mu}^T} \right\rangle_* = \left\langle \frac{\partial^2 E}{\partial\boldsymbol{\mu}\partial\boldsymbol{\mu}^T} \right\rangle_\mu = \Sigma^{-1}. \tag{2.99}$$

We also need to compute

$$\left\langle \frac{\partial E}{\partial\boldsymbol{\mu}}\frac{\partial E}{\partial\boldsymbol{\mu}^T} \right\rangle_\mu = \left\langle \Sigma^{-1}(\mu-\boldsymbol{x})(\mu-\boldsymbol{x})^T\Sigma^{-1} \right\rangle_\mu = \Sigma^{-1}\Sigma\Sigma^{-1} = \Sigma^{-1} \tag{2.100}$$

and since $\langle\partial E/\partial\mu\rangle_\mu = 0$ *we have*

$$\partial^2\Delta/\partial\boldsymbol{\mu}\partial\boldsymbol{\mu}^T = \Sigma^{-1} - 0 + \Sigma^{-1} - \Sigma^{-1} = \Sigma^{-1} \tag{2.101}$$

We may generalize this result to iterative learning situations where our current parameter estimate $\boldsymbol{\theta}$ *is close to a (local) maximum of the posterior distribution so that the neighborhood of the cost function becomes approximately quadratic. Once this happens, the terms* $\partial^2 E/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T$ *averaged with respect to the teacher and student distributions will cancel. The remaining terms are*

$$\mathbf{G} = \left\langle \frac{\partial E}{\partial\boldsymbol{\mu}}\frac{\partial E}{\partial\boldsymbol{\mu}^T} \right\rangle_\mu - \left\langle \frac{\partial E}{\partial\boldsymbol{\mu}} \right\rangle_\mu \left\langle \frac{\partial E}{\partial\boldsymbol{\mu}} \right\rangle_\mu \tag{2.102}$$

which is similar to the outer product approximation *of the Hessian matrix known from feed-forward neural network literature [32]. A 'cheap' learning rule for the parameter vector is achieved by using this matrix as an estimate of the second derivative for the Newton method*

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{k} + \mathbf{G}^{-1}\frac{\partial \Delta}{\partial \boldsymbol{\theta}} \tag{2.103}$$

■

2.5.4 Maximum Likelihood and Scale Parameters

Scale parameters are parameters that determine the order of magnitude of one or more state variables, and are as such typically strictly positive. For example, the variance of a Gaussian distribution σ^2 is a scale parameter for the Gaussian variable, since it determines the order of magnitude of the variation. A natural representation of a scale parameter is by its logarithm, so we may choose the alternative representation $\kappa = \log \sigma^2$. The choice of representation turns out to be of great importance for parameter estimation using the Boltzmann Learning rule. Recall that the Maximum Likelihood estimate corresponds to approximating $P(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{d}) = P(\boldsymbol{u} \mid \boldsymbol{\theta}, \boldsymbol{d}) P(\boldsymbol{\theta} \mid \boldsymbol{d})$ with $P(\boldsymbol{u} \mid \boldsymbol{d}, \boldsymbol{\theta})\delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{ML}})$ in the Bayesian integral (2.68) for finding the posterior distribution of the object image, given an observed noisy image. The exact location of the delta function peak is determined by the maximum likelihood criteria, and is therefore also directly dependent on the chosen representation.

Consider the scale parameter θ with posterior distribution $P(\theta \mid \boldsymbol{d})$ and the alternative representation $\kappa = \log \theta$ with corresponding posterior distribution $P(\kappa \mid \boldsymbol{d})$. Any mean with respect to these two distributions should be the same, i.e.

$$\int P(\theta \mid \boldsymbol{d}) q(\theta)\, d\theta = \int P(\kappa \mid \boldsymbol{d}) q(\theta)\, d\kappa \tag{2.104}$$

and we see that

$$P(\kappa \mid \boldsymbol{d}) = \frac{d\kappa}{d\theta} P(\theta \mid \boldsymbol{d}) \tag{2.105}$$

which ensures that the posterior distribution (2.68) is unchanged by the change of representation. For the log transform, we have $d\kappa / d\theta = d \log \theta / d\theta = 1/\theta$ so that

$$P(\kappa \mid \boldsymbol{d}) = \theta\, P(\theta \mid \boldsymbol{d}) \tag{2.106}$$

We identify the ML-estimates θ_{ML} and κ_{ML} as the maximum of the posterior distributions

$$\theta_{\text{ML}} = \arg \max_{\theta} P(\theta \mid \boldsymbol{d}) \qquad\qquad \kappa_{\text{ML}} = \arg \max_{\kappa} P(\kappa \mid \boldsymbol{d}), \tag{2.107}$$

but since in general $\kappa_{\text{ML}} \neq \log \theta_{\text{ML}}$ our ML-estimate will depend on whether we use $\kappa = \log \theta$ or θ as the native parameter representation.

The log-likelihood of the univariate Gaussian standard deviation σ is $-\log L(\sigma; x_{1\ldots N}) = N/\sigma - \sum x_i^2/2\sigma^2$ so that $\sigma_{\text{ML}} = \sum x_i^2/N$ while the maximum likelihood of the log-transformed σ is obtained by $L(\log \sigma; x_{1\ldots N}) = \sigma\, L(\sigma; x_{1\ldots N})$ and thus $(\log \sigma)_{\text{ML}} = \sum x_i^2/(N-1)$, which is also known as the central (unbiased) estimator of σ. The situation can be more subtle than a mere N vs. $N-1$ difference. Figure 2.9 illustrates a different situation. The left pane shows the posterior distribution of an un-bounded parameter. In this imaginary example, the ML estimate of the parameter is negative, and we will for now assume that this value of the parameter may lead to an unstable model. This is a similar to the unstable non-causal AR prediction models of Section 2.2 which are ML estimates (this is indicated in later Section 2.5.5), yet unstable. The shaded region is the part of parameter space of which we are sure that our model is stable, so we enforce the restriction that the parameter should be positive. The ML estimate within the leftmost shaded region is $\theta_{\text{ML}} \to 0^+$, which leads a poor approximation of the Bayesian integral (2.68). This estimate is found e.g. by iterating according to the learning rule

$$\theta^{(k+1)} = \theta^{(k)} - \eta \frac{\partial \Delta[P_*; P_\theta]}{\partial \theta} \tag{2.108}$$

Simply replacing θ with κ above produces the learning rule

$$\kappa^{(k+1)} = \kappa^{(k)} - \eta \frac{\partial \Delta[P_*; P_\theta]}{\partial \kappa} = \kappa^{(k)} - \eta\theta \frac{\partial \Delta[P_*; P_\theta]}{\partial \theta} \tag{2.109}$$

which corresponds to switching to a logarithmic abscissa of the posterior distribution (center pane of Figure 2.9), and does not help the situation; the ML-estimate is still $\kappa_{\mathrm{ML}} \to -\infty$. The log-transformed distribution $P(\kappa) = P(\log\theta)$ in the third pane is far better approximated with a single delta function located at the maximum value κ_{ML}.

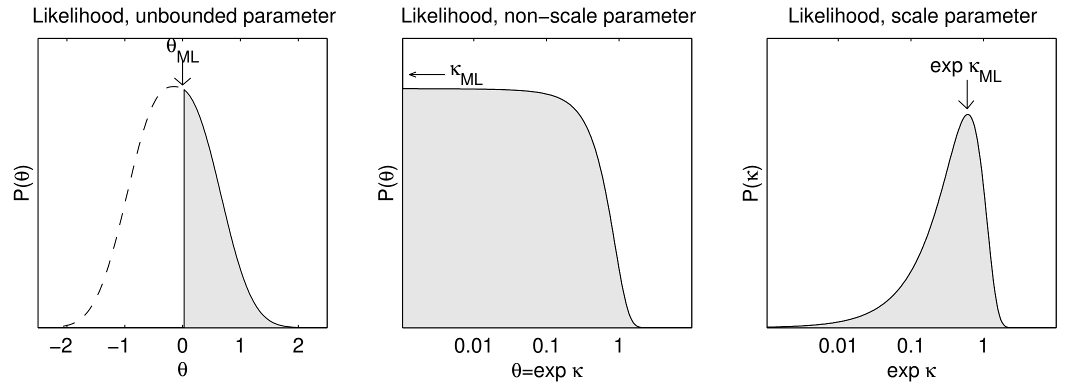


Figure 2.9: Leftmost pane shows a theoretical posterior distribution of a parameter which is restricted to be positive. The ML estimate is a bad estimate; reassigning the abscissa does not help the situation. The distribution needs to be transformed to get a ML-estimator which approximates the posterior distribution better.

Energy functions in Gibbs distributions are often designed as linear combinations of base terms

$$E(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_i \theta_i \, \varepsilon_i(\boldsymbol{x}; \boldsymbol{\theta}) \tag{2.110}$$

The requirement that $E(\boldsymbol{x}; \boldsymbol{\theta})$ is bounded from below (finite partition function) is usually synonymous with the requirement $\theta_i > 0$ for all i. This argues that the parameters θ_i are actually scale parameters, and should be treated as such in the learning process. In the following it will be described how the Boltzmann learning rule can be modified to take account of scale parameters changing only the updating rule of the learning algorithm.

The Modified Learning Rule for Scale Parameters

Writing out explicitly the conditional distribution of the transformed parameter vector given the example $\boldsymbol{d}$, we have in general for a K-dimensional parameter vector[6]

$$P_\theta(\kappa_{\mathrm{i}} \,|\, \boldsymbol{d}) = \Pi_{k=1}^{(k)} \theta_{\mathrm{k}} \, P_\theta(\theta_{\mathrm{i}} \,|\, \boldsymbol{d}), \qquad \kappa_{\mathrm{i}} = \log\theta_{\mathrm{i}} \tag{2.111}$$

and using Bayes relation we readily get

$$P_\kappa(\boldsymbol{\kappa} \,|\, \boldsymbol{d}) = \Pi_{k=1}^{(k)} \theta_{\mathrm{k}} \, P_\theta(\boldsymbol{\theta}) P_\theta(\boldsymbol{d} \,|\, \boldsymbol{\theta}) \tag{2.112}$$

which indicates that the change of representation can be implemented by a (degenerate) prior $\Pi_{k=1}^{(k)} \theta_{\mathrm{k}}$ into the Boltzmann derivatives (2.75) and (2.76) in line with (2.91) and (2.91)

$$\frac{\partial \Delta[P_*, P_\kappa]}{\partial \theta_{\mathrm{i}}} = -\frac{\partial P_\kappa(\kappa \,|\, \boldsymbol{d})}{\partial \theta_{\mathrm{i}}} = \frac{\partial \Delta[P_*, P_\theta]}{\partial \theta_{\mathrm{i}}} - \frac{\partial \log \Pi_k \theta_{\mathrm{k}}}{\partial \theta_{\mathrm{i}}} = \frac{\partial \Delta[P_*, P_\theta]}{\partial \theta_{\mathrm{i}}} - \frac{1}{\theta_{\mathrm{i}}} \tag{2.113}$$

and similarly for the second derivative, which is affected only on the diagonal

$$\frac{\partial^2 \Delta[P_*, P_\kappa]}{\partial \theta_{\mathrm{i}} \partial \theta_{\mathrm{j}}} = -\frac{\partial^2 P_\kappa(\kappa \,|\, \boldsymbol{d})}{\partial \theta_{\mathrm{i}} \partial \theta_{\mathrm{j}}} = \frac{\partial^2 \Delta[P_*, P_\theta]}{\partial \theta_{\mathrm{i}} \partial \theta_{\mathrm{j}}} + \frac{\delta_{i-j}}{\theta_{\mathrm{i}}^2} \tag{2.114}$$

[6] We define $\exp \boldsymbol{\kappa} = [\exp(\kappa_1), \ldots, \exp(\kappa_{\mathrm{K}})]$.

The learning algorithm can be aided by performing the iterations in the transformed parameter $\boldsymbol{\kappa}$. By doing this, we ensure that the stepping algorithm never produces negative parameter values by accident if, for example, the step size was too large. To do this, we need to change the derivatives with respect to $\boldsymbol{\theta}$ into derivatives with respect to $\boldsymbol{\kappa}$. By adopting the chain rule we have

$$\frac{\partial \Delta}{\partial \boldsymbol{\kappa}} = \frac{\partial \Delta}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\kappa}} = \boldsymbol{\theta} \odot \frac{\partial \Delta}{\partial \boldsymbol{\theta}} \tag{2.115}$$

where we have used the symbol $\odot$ for element by element multiplication (like the Matlab "`.*`" operator). The gradient descent algorithm translates into

$$\boldsymbol{\kappa}^{(k+1)} = \boldsymbol{\kappa}^{(k)} - \eta \frac{\partial \Delta[P_*, P_\kappa]}{\partial \boldsymbol{\kappa}} \tag{2.116}$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} \exp\big(-\eta \boldsymbol{\theta} \odot \frac{\partial \Delta[P_*, P_\kappa]}{\partial \boldsymbol{\theta}}\big) = \boldsymbol{\theta}^{(k)} \exp\big(-\eta(\boldsymbol{\theta} \odot \frac{\partial \Delta[P_*, P_\theta]}{\partial \boldsymbol{\theta}} - 1)\big) \tag{2.117}$$

Second order learning is also possible. First we note that

$$\frac{\partial \Delta^2}{\partial \kappa_i \partial \kappa_j} = \frac{\partial}{\partial \kappa_j}\Big(\frac{\partial \Delta}{\partial \theta_i}\frac{\partial \theta_i}{\partial \kappa_i}\Big) = \theta_i \frac{\partial}{\partial \kappa_j}\frac{\partial \Delta}{\partial \theta_i} = \theta_i \frac{\partial \theta_j}{\partial \kappa_j}\frac{\partial^2 \Delta}{\partial \theta_i \partial \theta_j} = \theta_i \theta_j \frac{\partial^2 \Delta}{\partial \theta_i \partial \theta_j} = \boldsymbol{\theta}\boldsymbol{\theta}^T \odot \frac{\partial^2 \Delta}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \tag{2.118}$$

The Newton iterations are thus performed with the steps

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} \exp\left(-\eta \left[\boldsymbol{\theta}\boldsymbol{\theta}^T \odot \frac{\partial^2 \Delta[P_*, P_\kappa]}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}\right]^{-1} \cdot \big(\boldsymbol{\theta} \odot \frac{\partial \Delta[P_*, P_\kappa]}{\partial \boldsymbol{\theta}}\big) \right), \tag{2.119}$$

where $\partial \Delta[P_*, P_\kappa]/\partial \boldsymbol{\theta}$ and $\partial^2 \Delta[P_*, P_\kappa]/\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T$ are found using (2.113) and (2.114) respectively.

It is possible, of course, for scale parameters to co-exist in a parameter vector with non scale parameters. First order learning (i.e. gradient descent) of a mixed parameter vector is achieved by simply using (2.117) for the scale parameters and the usual gradient descent rule for the remaining parameters. Second order learning is slightly more complex but straightforward, and is not described here.

Learning the Covariance Matrix of a Multivariate Gaussian Distribution

We shall derive the first order derivatives for the learning of the covariance matrix in Gaussian distributions. Due to the large number of elements in the covariance matrix we will assume a parametric form $\mathbf{Q} = \mathbf{Q}_\theta$, and we will furthermore assume that all variables are zero mean. We note that the Gaussian covariance matrix needs to be positive definite in order for the distribution not to be degenerate. This can be ensured by designing the parameterization $\mathbf{Q}_\theta$ as a linear combination with positive coefficients of positive definite basis matrices $\mathbf{Q}_\theta = \sum_k \theta_k \mathbf{K}_k$ and letting the θ_k be scale parameters. Alternatively, the parameterization can be in the inverse $\mathbf{Q}_\theta^{-1} = \sum_k \theta_k \mathbf{K}_k$.

A typical degeneration model for images is the additive noise model: The observed image is modeled as an object image plus a noise image. In terms of the covariance matrix, this means that $\mathbf{Q}$ is the sum of the noise covariance and object covariance $\mathbf{Q}_\theta = \Sigma_N(\boldsymbol{\theta}) + \Sigma_u(\boldsymbol{\theta})$. The Boltzmann learning rule can be applied to the learning of the parameter vector $\boldsymbol{\theta}$ given example images. With the matrix $\mathbf{Q}$ separated into $\Sigma_N(\boldsymbol{\theta}) + \Sigma_u(\boldsymbol{\theta})$ in hand, the remaining task of finding the most probable object $\boldsymbol{u}$ given the observed image $\boldsymbol{d}$ is a Wiener filtering problem (as described in Section 2.3), and can be solved by direct inversion of the covariance equations (2.40).

A requirement for the resulting image restoration to succeed is that the object and noise parametric covariances are adequately different. What exactly 'adequately' implies is not easily expressed formally. Say, for example, that they are equal: $\Sigma_N = \theta_1 \mathbf{K}$ and $\Sigma_u = \theta_2 \mathbf{K}$. Adopting a learning scheme for $\boldsymbol{\theta} = (\theta_1, \theta_2)$ would lead to well determination of the global level $\theta_1 + \theta_2$ while the individual θ_1, θ_2 are indeterminable. As a result, the posterior distribution of parameters is highly correlated. Recalling the Wiener filter matrix (2.40) for restoration of the object signal $\mathbf{G} = \big(\theta_1 \Sigma_N + \theta_2 \Sigma_u\big)^{-1} \theta_2 \Sigma_u$, which in this simple example becomes the identity matrix multiplied with the constant $G = \theta_2/(\theta_1 + \theta_2)$. Hence, all values for G ranging from 0 to 1 are equally probable. This example demonstrates that dependencies between noise covariance parameters and object covariance parameters is an indicator for 'inadequately different' noise and object covariance models. It may be just a problem of choosing a different parameterization of the signals. Of course, if the restoration problem is such that the noise and object do indeed have very similar statistical properties, we have a more serious problem. We then need to estimate the noise and object statistics separately using other methods prior to the restoration. But even with correct—while similar in correlation structure—noise and object model covariance matrices present, the resulting Wiener filter matrix will do only a very small amount of noise reduction to the observed image.

To derive the Kullbak derivatives, we first note the energy function of the zero mean multivariate Gaussian distribution

$$E(\boldsymbol{d};\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{d}^T\mathbf{Q}_\theta^{-1}\boldsymbol{d} \tag{2.120}$$

and using the rule for matrix differentiation $\frac{\partial \mathbf{Q}_\theta^{-1}}{\partial \boldsymbol{\theta}} = -\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}$

$$\frac{\partial E(\boldsymbol{d};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\boldsymbol{d}^T\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}\boldsymbol{d} \tag{2.121}$$

The mean with respect to the student distribution is found using the result for Gaussian distributions $\langle \boldsymbol{y}^T\mathbf{A}\boldsymbol{y}\rangle = \mathrm{Tr}\,[\mathbf{A}\Sigma_y] + \langle \boldsymbol{y}^T\rangle\mathbf{A}\langle \boldsymbol{y}\rangle$ where Σ_y is the covariance matrix of $\boldsymbol{y}$. In our case we consider zero mean signals so we get

$$\left\langle \frac{\partial E(\boldsymbol{d};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_\theta = -\mathrm{Tr}\left[\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}\mathbf{Q}_\theta\right] \tag{2.122}$$

The teacher average is found by summation over teacher examples $\boldsymbol{d}_{(1)}\ldots\boldsymbol{d}_{(P)}$

$$\left\langle \frac{\partial E(\boldsymbol{d};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle_* = -\frac{1}{P}\sum_{p=1}^{P}\boldsymbol{d}_{(p)}^T\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}\boldsymbol{d}_{(p)} \tag{2.123}$$

Defining $\mathbf{D} = [d_{ip}]$ as the matrix of example vectors $\boldsymbol{d}_{(p)}$ we have $\sum_p \boldsymbol{d}_{(p)}^T\mathbf{A}\boldsymbol{d}_{(p)} = \sum_{p,i,j} d_{(p),i}a_{ij}d_{(p),j} = \mathrm{Tr}\,[\mathbf{D}^T\mathbf{A}\mathbf{D}] = \mathrm{Tr}\,[\mathbf{A}\mathbf{D}\mathbf{D}^T] = P\mathrm{Tr}\,[\mathbf{A}\Sigma_*]$ where $\Sigma_* = \mathbf{D}\mathbf{D}^T/P$ is the empirical covariance estimate for the teacher examples. Combining this result with (2.123) and (2.122) we have

$$\frac{\partial \Delta}{\partial \boldsymbol{\theta}} = \mathrm{Tr}\left[\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}(\mathbf{Q}_\theta - \mathbf{Q}_*)\right] \tag{2.124}$$

The main problem with direct practical application of the above equation is the computational requirement for large example vectors. For each iteration of the learning scheme we need to invert matrices which are $N \times N$ for vector lengths of N. For example, learning the covariance structure of a $10 \times 10 \times 10$ block of voxels requires inverting a 1000×1000 sized matrix; a numerically very unstable procedure. Having many example vectors does not increase the computational complexity. Inversion of an $N \times N$ symmetrical and positive definite matrix using Cholesky factorization requires the order of $\mathcal{O}(N^3)/6$ operations [8, Ch. 2.9]. This enforces a computational wall on the estimation problem, which is prohibitive for the practical implementation of algorithms based on (2.124). If the example vectors are too large, we need to break up the examples into smaller pieces using window functions. This may be problematic if the correlations of the signal have a larger spatial distribution than the size of the window, in which case a coarser sampling of the image could be used. An alternative is to switch to separable models,

A weighting function can be implemented into (2.124) simply by introducing the diagonal weighting matrix $\mathbf{W}$

$$\frac{\partial \Delta}{\partial \boldsymbol{\theta}} = \mathrm{Tr}\left[\mathbf{Q}_\theta^{-1}\frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}}\mathbf{Q}_\theta^{-1}\mathbf{W}(\mathbf{Q}_\theta - \mathbf{Q}_*)\mathbf{W}\right] \tag{2.125}$$

It can be advantageous to use a parametric form of the covariance matrix into a noise and object covariance matrix as discussed earlier. This parametric form can be chosen so that the actual application of the Wiener filter matrix can be performed as a mean field annealing process in a Markov random field. Recall that the mean field iterations in the Markov Random Field were designed to iteratively solve for the MAP estimate $\boldsymbol{u}_{\mathrm{ML}} = \arg \underset{\boldsymbol{u}}{\max}\, P(\boldsymbol{u}|\boldsymbol{d})$ by iteratively minimizing the cost function $E(\boldsymbol{u},\boldsymbol{d};\boldsymbol{\theta})$.

This means that we may estimate the image parameters in a small window using the above windowed learning rule, and subsequently apply the Wiener filter on the entire image using the mean field method which scales computationally with $\mathcal{O}(N)$. We will demonstrate this approach with a small example.

Example 2.6 Parameter estimation in Gaussian Random Fields with predetermined edges. *We shall illustrate the use of the learning of parameterized Gaussian covariance matrices on an MRI slice of the human brain. We will investigate the estimation of parameters in random fields that live in 'bounded' areas, like e.g. brain signals. The bounds are implemented into the energy function by the edge map*[7] $\boldsymbol{p}$ *and* $\boldsymbol{q}$*, similar to (2.56) in Example 2.3*

$$E_{prior}(\boldsymbol{u},\boldsymbol{p},\boldsymbol{q},\boldsymbol{r}) = \frac{\nu}{2}\sum_{ij}\Big[(1-p_{ij})(u_{ij}-u_{i+1,j})^2 + (1-q_{ij})(u_{ij}-u_{i,j+1})^2\Big] \tag{2.126}$$

with the alteration that now the edges are assumed to be known in advance. The edges are acquired by e.g. a segmentation of the MR slice into tissue types as shown in Figure 2.10. Since the matrix inversion is computationally demanding, we will in this example examine the signal structure of gray matter inside a 2D slice of 21×21 *voxels.*

The segmentation was obtained by applying a linear (Gaussian) classifier to manually co-registered sinc-resampled t1-weighted (MPRAGE) and t2-weighted (FISP) MRI scans of the same subject, in $1\,mm^3$ *sized voxels. Prior to the threshold, both volumes were smoothed using a small sized (Full-Width-Half-Max* $\simeq 2\,mm$ *) Gaussian blurring kernel. The thresholds were determined from a set of manually identified voxels. Each tissue class was represented with a two dimensional Gaussian distribution with local mean and covariance and the voxels were classified according to the class with maximum posterior probability.*

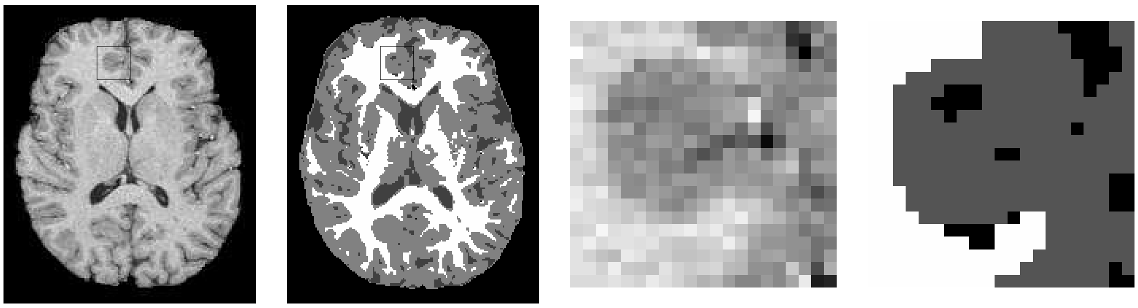


Figure 2.10: The selected MRI window of 21 × 21 pixels, together with the segmentation into white matter (light gray), gray matter (darker gray), CSF (darkest shade) and background to use for the parameter estimation experiment.

We assume an additive noise model, so that we have the parameterization $\mathbf{Q} = \nu\Sigma_u + \omega\Sigma_N$*, where the parameters* ν *and* ω *are positive scale parameters. Positive parameter values is a sufficient condition for the model to be stable. This can be seen by inspecting the second derivative* $\partial^2 E_{posterior}/\partial u_{i,j}^2 = \nu \cdot (p_{i,j} + p_{i-1,j} + q_{i,j} + q_{i,j-1}) + \omega$ *which is positive for positive parameter values. It is not a necessary condition—stable models do exist for negative parameter values only these are treated here as 'special cases'.*

The line sites p_{ij} *and* q_{ij} *are active when the tissue type of voxel* (i,j) *is different from* $(i+1,j)$ *and* $(i,j+1)$ *respectively. This information enters the object covariance matrix* Σ_u *so that the expression*

$$-\log[P(\boldsymbol{u})P(\boldsymbol{d}|\boldsymbol{u})] = \frac{\nu}{2}\boldsymbol{u}^T\Sigma_u^{-1}\boldsymbol{u} + \frac{\omega}{2}(\boldsymbol{u}-\boldsymbol{d})^T\Sigma_N^{-1}(\boldsymbol{u}-\boldsymbol{d}) \tag{2.127}$$

equals the energy function (2.126). This is achieved as shown in Figure 2.11.

We are able to compute the posterior probability numerically by direct evaluation of the Gaussian normalization constant $L(\nu,\omega;\boldsymbol{d}) = \sqrt{(2\pi)^N|\Sigma_d|}\exp(-\frac{1}{2}\boldsymbol{d}^T\Sigma_d^{-1}\boldsymbol{d})$ *and similarly for the* log-*transformed* $L(\log\nu,\log\omega;\boldsymbol{d}) = \nu\omega L(\nu,\omega;\boldsymbol{d})$*. The likelihood is shown together with the Kullbak derivatives in Figure 2.12 for both representations.*

[7]Note that we have redefined the values of the edge elements: $p_{i,j} = 0/1$ now corresponds to active/inactive edge element.

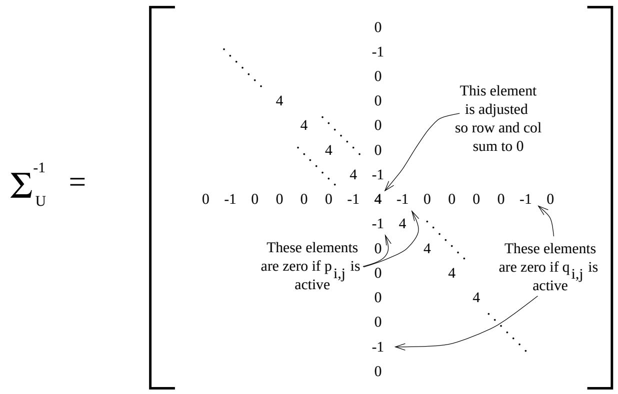


Figure 2.11: Construction of the inverse object object covariance matrix from a predetermined edge map.

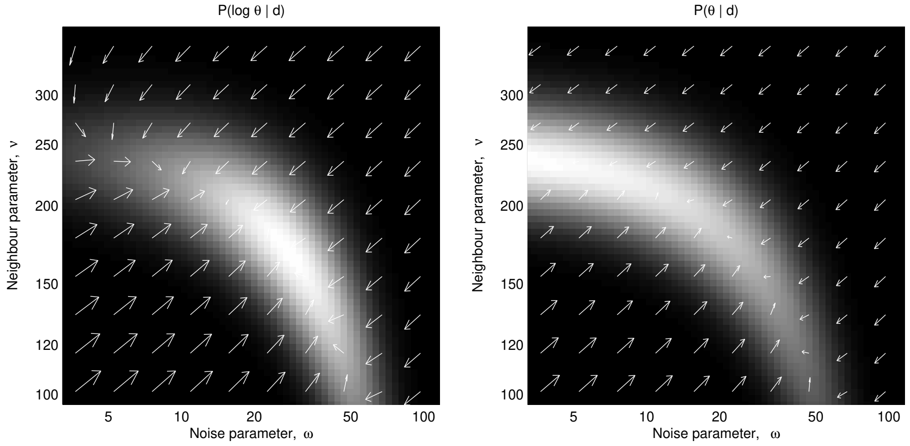


Figure 2.12: The posterior distribution of parameters for a Gaussian Markov Random field with predetermined edges. The log-transformed parameter has a more compact distribution which is better modeled by the ML estimate. The two images are computed for comparison using the same logarithmic span of parameter values; therefore the right distribution is not shown in its 'native' linear coordinate system (like the middle pane of Figure 2.9).

It is evident that the log-transformed parameter in this case has a more compact distribution, which is better modeled by the ML estimate. Simply following the non-scale gradient causes ω to vanish $\omega \to 0$, while $\log \omega$ attains stable values.

■

2.5.5 Estimation of Non-causal AR Models Using the Boltzmann Learning Rule

We will now consider the generalization of the Boltzmann Learning rule for stationary signals. We will use the learning rule for covariance matrices (2.124) now assuming that the matrices are of infinite size and have Toeplitz structure similar to the considerations in Section 2.2.2.

As noted earlier in Section 2.4.1, the AR model (which for simplicity will be assumed two dimensional) can be said to have the parameterization

$$\mathbf{Q}^{-1} = \frac{1}{\beta^2} \begin{bmatrix} \ddots & \ddots & \ddots & & & & \\ \ddots & 1 & -a_1 & -a_2 & & & \\ \ddots & -a_1 & 1 & -a_1 & -a_2 & & \\ & -a_2 & -a_1 & 1 & -a_1 & -a_2 & \\ & & -a_2 & -a_1 & 1 & -a_1 & \ddots \\ & & & -a_2 & -a_1 & 1 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{bmatrix} \tag{2.128}$$

with some mapping of the parameters $a_{p,q}$ into the diagonals of $\mathbf{Q}^{-1}$. The derivatives of the AR model parameters can be found by

$$\frac{\partial \Delta}{\partial \boldsymbol{\theta}} = \mathrm{Tr}\left[\mathbf{Q}_\theta^{-1} \frac{\partial \mathbf{Q}_\theta}{\partial \boldsymbol{\theta}} \mathbf{Q}_\theta^{-1} (\mathbf{Q}_\theta - \mathbf{Q}_*)\right] \tag{2.129}$$

As has been shown earlier, matrix multiplication of Toeplitz matrices equals convolution of the signals stored in the matrix columns (rows) at the limit of infinite matrix size. Designating the inverse covariance "signal" $R_\theta^-(p,q)$ in line with Section 2.2.2 we translate the above equation into

$$\frac{\partial \Delta}{\partial \theta} = \left(R_\theta^- * \frac{\partial R_\theta}{\partial \theta} * R_\theta^- * (R_\theta - R_*)\right)(0,0) \tag{2.130}$$

The trace operation again translates into sampling the convolution result at (0,0) since this is mapped to the diagonal. We see that the above equation corresponds to the Fourier transform

$$\frac{\partial \Delta}{\partial \theta} = \int_{-\pi}^{\pi} \frac{1}{S_\theta^2} \frac{\partial S_\theta}{\partial \theta} (S_\theta - S_*)\, d\omega \tag{2.131}$$

replacing the convolution operator with multiplication. By inserting the non-causal AR model spectrum $S_\theta = \beta^2 / A(z_1, z_2, z_3)$, $A(z_1, z_2, z_3) = 1 - \sum_{p,q \in \mathcal{W}} a(p,q) z_1^{-p} z_2^{-q}$ we have

$$\frac{\partial S_\theta}{\partial a_{p,q}} = -\frac{\beta^2}{A^2} \frac{\partial A}{\partial a_{p,q}} = \frac{S_\theta^2}{\beta^2} z_1^{-p} z_2^{-q} \tag{2.132}$$

so that

$$\frac{\partial \Delta}{\partial a_{p,q}} = \frac{1}{\beta^2} \int_{-\pi}^{\pi} (S_\theta - S_*) z_1^{-p} z_2^{-q}\, d\omega \tag{2.133}$$

which is identical to the Fourier transform of the error of the spectral fit $S_\theta - S_*$. The noise level is obtained from the derivative

$$\frac{\partial \Delta}{\partial \beta^2} = \frac{1}{\beta^2} \int_{-\pi}^{\pi} \left(1 - \frac{A(\omega) S_*}{\beta^2}\right) d\omega \tag{2.134}$$

which can be solved by setting $\beta^2 = \int_{-\pi}^{\pi} A(\omega) S_*\, d\omega / (2\pi)^p$ where p is the number of dimensions in the signal.

We may construct a learning scheme for the estimation of AR model parameters using the Boltzmann Learning Rule on the above result. We see that the gradient with respect to the AR model parameters is obtained simultaneously using a single inverse Fourier transform of the error spectrum. The observed spectrum S_* can be approximated by a discrete Fourier transform of zero padded data. The above inverse Fourier integral can be approximated by an inverse FFT, picking out the central (low frequency) part of the inverse transformed spectrum for the gradients.

The above learning rule is capable of fitting AR models *provided the AR model remains stable* during the training iterations. If the current model parameters dictates an unstable AR model, S_θ will diverge as will the gradient. This is most likely to happen if the AR model obtained from solving the normal equations (2.11, 2.12) is not stable.

This experience tells us that maximizing the likelihood, which is the basis of the Boltzmann Learning Rule, is not always the optimal optimization criteria when areas of parameter space yields unstable models.

Is special care is not taken, nothing will prevent the parameters to iterate into these areas. For parameter estimation in Gaussian random fields, one should disallow combinations of parameter values which yields unstable models.

A more general method for ensuring stability is by enforcing the restriction that the model parameters should be positive, and model the covariance matrix (or, alternatively the inverse covariance matrix) using linear combinations with positive parameters of known, stable basis models, i.e.

$$\mathbf{Q}^{(-)} = \sum_i \mathbf{K}_i \theta_i \tag{2.135}$$

and letting the θ_i be scale constants. Parameterizing $\mathbf{Q}$ in this manner corresponds to fitting linear combinations of Moving Average (MA) models, while the modeling of $\mathbf{Q}^{-1}$ corresponds to fitting linear combinations of AR models. Each base AR model $\mathbf{K}_i$ should, of course, be known stable.

2.6 Concluding Remarks

We have presented the general theory for the use of Markov Random Fields in image modeling. It has been shown how the general linear theory of AR model estimation can be linked to Markov random field parameter estimation. Furthermore, Markov Chain methods have been described with the special application to image restoration of brain image volumes. Mean field annealing has been demonstrated as an alternative to solving the Wiener system equations, which is computationally prohibitive for large images.

The inherent stability problems of the linear theory has been shown to directly affect the stability of parameter estimation in Markov random fields. The use of log-transformed parameters has been proposed as a workaround solution to the AR stability problem. More work is required in this respect in order to fully clarify the connection between the solution of the AR normal equations and the ML estimate.

Chapter 3

Non-linear Image Registration

This chapter describes a novel method which has been developed for performing non-linear normalization of MRI scans of human brains. The method has been applied to a functional $[^{15}O]$water PET study of saccadic eye movements. This experiment and warp algorithm is described in detail in the following sections 3.1–3.6, which comprise a manuscript submitted to the IEEE Transactions in Medical Imaging. The algorithm is an extension of the algorithm described in a previous publication (see Appendix B, pages 78–88) with the inclusion of different voxel similarity measures to use for optimizing the warp displacement field, including correlation coefficient and mutual information. Since the submission of the aforementioned manuscript, a novel extension has been added; namely the use of *reconciled* warps. This extension is described in further detail in Section 3.8.

The programs for non-linear registration have been made publicly available. Documentation for the use of the program is reproduced in Appendix B.

3.1 A Non-linear Brain MRI Registration Algorithm

When investigating functional activation patterns in multiple subjects, a basic problem is that of comparing and averaging the functional activation images across subjects. Apart from rigid body translation and rotation, the inter-subject registration problem is essentially how to find a three dimensional deformation of a subject's anatomy to match a template anatomy. The deformation ideally represents a *correspondence map*, between the observed image and the template image, to the extent that such a map exists.

Multiple scans from one or more sensor modalities of the same subject may be automatically registered quite accurately using rigid body rotations and translations, involving 6 parameters in the transformation procedure. In [33] it is shown how intra-subject registration MRI-MRI and PET-MRI can be achieved with good accuracy across the entire volume ($\sim$1–2 mm) using the Automated Image Registration (AIR) method as described in [5, 6, 34, 35].

In contrast to this, inter-subject registration is more complex in terms of the number of parameters necessary: Ultimately, the correspondence map consists of three parameters for each voxel, representing the x, y and z location of the corresponding voxel in a template image typically defined in the standard Talairach stereo-tactic space [36]. Automated inter-subject registration procedures typically overcome this parameter estimation problem by parameterizing the deformation field in a lower dimension. An example of this is the 12-parameter transformation supported by the AIR package [5, 6, 34, 35] which recently has been extended to include 5th order polynomial transformations with up to 168 parameters.

This paper compares the resulting functional activation patterns achieved when applying non-linear registration techniques across subjects to those achieved using conventional 12-parameter affine[1] transformations. The inter-subject functional variation (e.g. the variance across subjects in $[^{15}O]$water PET studies) is composed of a mixture of measurement noise, anatomical variation and functional variation, e.g. variation in how a given subject's activation pattern is linked to the anatomy. Since the inter-subject structural registration is limited to a certain spatial scale, we need to choose the degree to which the structural registration should be performed. The image resolution together with the functional variation between subjects set the lower boundary on the scale at which the anatomical registration will be useful. It may be that the effort of

[1] An affine transformation is one that preserves straight parallel lines. It is conveniently expressed as a matrix multiplication of a 4 by 4 transformation matrix with a homogeneous coordinate vector [37].

performing a detailed non-linear structural registration is not rewarded by an increase in the extracted functional activation signal, because the non-linear registration operates at a smaller scale than the functional differences that are known to exist within a set of normal subjects. We wish to know whether the proposed non-linear registration algorithm leads to an "improvement" in the measured brain activity map, as opposed to using a simpler conventional method, e.g. using 12-parameter affine transformations.

In the non-linear registration procedure presented different voxel similarity measures may be applied and it is not evident beforehand which of these measures is the optimal. Furthermore, the algorithm uses a set of regularizing parameters, which control the smoothness of the displacement field that is obtained, and we need to consider how to find an optimal setting for such parameters.

In order to address the above problems, we have evaluated the anatomical registration by examining how well the corresponding functional data have been registered and how the functional activation signal is improved. It is assumed that the optimal registration is the one that makes the functional activation signal stand out most clearly from the baseline condition for a group of subjects. Using this evaluation approach we are able to compare different registration techniques, voxel similarity measures and regularization parameter settings.

The rest of this chapter is arranged as follows. The non-linear registration algorithm is described in Section 3.2. In Section 3.3 we describe the functional data set used for the alignment experiments, and in Section 3.4 the analysis methods used for examining the functional signal space are described and the results of the analysis are discussed.

3.2 The Non-linear Registration Algorithm

The method described here is based on a 3D extension of the approach described in [38] although several extensions have been added. The method is also similar in construction to the ANIMAL algorithm developed by Collins *et al.* [39, 40] and the methods described in [41, 42]. A considerable amount of work exists based on non-linear deformations of human brain anatomy—most of the work has been done with the approach of investigating the anatomical variation between individuals [43, 44]. But few authors have investigated to what extent functional inter-subject variation in brain volumes can be corrected for using the anatomical information [45].

The major difference between this method and previously published methods lies in the regularization applied and in the evaluation of the voxel similarity measures. Instead of optimizing the similarity measure locally, this method performs the optimization based on the change in the global similarity measure subject to regularizing constraints. Furthermore, the method is optimized to yield fast convergence, so that a full non-linear registration of a high-resolution $256 \times 256 \times 160$ volume with a $64 \times 64 \times 40$ vector field can be computed in 25 minutes on a 200 MHz R4400 processor. The techniques for evaluation of global change of similarity measures presented below may also be adapted to warp-algorithms that apply different strategies for searching for the optimal displacement field, e.g. the wavelet based method of [46].

A volume of displacement vectors Φ overlaying the template volume is computed. These vectors are sampled on a (typically 4 times) coarser grid and represent local displacements that make the subject image volume similar to the template image. The vector field is computed in a hierarchical manner, by first iterating the displacement field using a coarsely sampled vector grid on sub-sampled instances of the subject and template volumes, and increasing resolution whenever the similarity measure has converged until the original resolution of the MR volumes is reached. Once the set of displacement vectors is computed, the actual displacement of each voxel in the high-resolution volume may be found by tri-linear interpolation between the 8 local surrounding vectors on the (coarser sampled) grid of displacement vectors.

3.2.1 The Algorithm

The details of the algorithm are explained using a pseudo-C description in Table 3.1 . We have had stable results when the vector field is four times coarser than the template image, i.e. each vector is associated with the $4 \times 4 \times 4$ cube of nearest voxels. Three resolution steps have been sufficient for all scans tested when the scans have been pre-aligned using a 12-parameter affine transformation prior to the non-linear warp. We have used our own implementation of the AIR method [33, 6] for this step.

Let $\mathcal{A}(\boldsymbol{v})$ and $\mathcal{B}(\boldsymbol{v})$ be the template and subject image volumes, respectively, indexed by the vector $\boldsymbol{v} = (p, q, r) \in \mathcal{G}$, consisting of discretely sampled and (8 bit integer) quantized voxel intensities a and b. The deformation field at iteration n, denoted Φ^n, consists of vectors Φ^n_{ijk} with the coarser grid indexed by

(i, j, k). The template voxel $\mathcal{A}(\boldsymbol{v})$ is subject to a shift in coordinates so that $\mathcal{A}(\boldsymbol{v})$ is mapped to $\mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}})$, where $\Phi^n_{\boldsymbol{v}}$ is obtained by tri-linear interpolation of the 8 nearest displacement vectors on the coarser sampled grid of vectors Φ^n_{ijk} as shown in Figure 3.1.

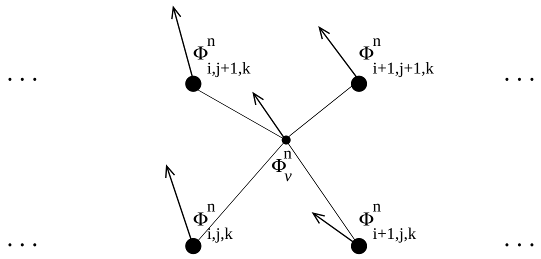


Figure 3.1: Interpolation of displacement field vectors. The notation $\Phi^n_{\boldsymbol{v}}$ implies a tri-linear interpolation of the 8 surrounding grid vectors Φ^n_{ijk} to find the displacement for the voxel at location $\boldsymbol{v}$.

Different assumptions on the exact nature of this mapping may be made depending on the sources of image volumes $\mathcal{A}$ and $\mathcal{B}$. If the scans are recorded using identical acquisition sequences on the same scanner, it may be fair to assume a linear mapping of the voxel intensities e.g. the correlation voxel similarity measure. Such an approach may not be viable if the tissue contrasts differ. In such cases a more general mapping should be assumed—e.g. the mutual information (MI) similarity measure. MI has gathered considerable interest for inter-modality image registration, see e.g. [47, 48, 49, 50, 51]. Variations in tissue contrast *between* the volumes should not be confused with variation *within* a single volume typically caused by field inhomogeneities—such effects are presumed to have been corrected for prior to the registration process.

`for resolution = quarter, half, full do`

> subsample the template and subject volumes to current resolution using Hann weighted sinc sub-sampling kernel.
>
> Interpolate displacement field so that vectors exist for each $4 \times 4 \times 4$ block of voxels in the template image.
>
> repeat
>
> > Compute cost function look-up table $\mathcal{T}(a, b)$ from current histogram $h^n(a, b)$.
> >
> > Update elements Φ_{ijk} for which $i+j+k$ is odd maximizing the estimated decrease in the global cost function
> >
> > $$\Phi^n_{ijk} = \underset{\phi \in S_{ijk}}{\text{minarg}} \; \Delta\hat{C}_\phi$$
> >
> > Update elements Φ_{ijk} for which $i+j+k$ is even.
> >
> > Apply low-pass filter to Φ^n.
> >
> > Compute new histogram of matched image intensities $h(a, b)$, and compute the global similarity measures from this.
>
> until (selected similarity measure has converged)

`end for`

Table 3.1: The outline of the warping algorithm.

Iteration Procedure

At each iteration step, vector estimates are updated by searching for the largest possible estimated decrease in a global cost function, which is assumed to take the form

$$C(\Phi^n) = D\{\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}})\}_{\boldsymbol{v}\in\mathcal{G}} + R(\Phi^n), \tag{3.1}$$

with $D(a,b)$ being a function measuring the similarity of a set of matched voxel intensities a and b, and $R(\Phi^n)$ is a regularizing constraint representing our prior knowledge of the nature of the displacement field, i.e. that it is smooth. We have applied a simple Gaussian Markov random field [18] prior consisting of the sum of squared differences of the 6 nearest neighboring vectors indexed by $i'j'k' \in \mathcal{N}_{ijk}$

$$R(\Phi^n) = \frac{\alpha}{2} \textstyle\sum_{ijk} \sum_{i'j'k'\in\mathcal{N}_{ijk}} \left\| \Phi^n_{ijk} - \Phi^n_{i'j'k'} \right\|^2, \tag{3.2}$$

introducing the global regularization parameter α. In the following we shall assume that D can be expressed as a function of the probability distribution of matched voxel intensities $p(a,b)$. The estimated change of $C(\Phi^n)$ from moving a single displacement vector Φ^n_{ijk} to ϕ is computed by a linear approximation. The notation $C^n_{\phi,ijk}$ denotes the value of the cost function (3.1) when replacing the single vector Φ^n_{ijk} by ϕ.

$$\Delta C^n_{\phi,ijk} \simeq \Delta R(\Phi^n) + \sum_{a,b} \Delta p(a,b) \frac{d}{dp(a,b)} D(\Phi^n). \tag{3.3}$$

The distribution $p(a,b)$ may be estimated directly from the two-dimensional discrete histogram $h^n(a,b)$ given the current estimate of the displacement field:

$$h^n(a,b) = \sum_{\boldsymbol{v}\in\mathcal{G}} \delta\{\mathcal{A}(\boldsymbol{v}) - a, \mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}}) - b\} \tag{3.4}$$

$$p(a,b) \simeq h^n(a,b)/N, \qquad N = \sum_{a,b} h^n(a,b) \tag{3.5}$$

A limited neighborhood centered around the previous iteration vector is searched exhaustively evaluating equation (3.3) at integer displacements of the corresponding block of template voxels. The neighborhood is limited to ± 2 voxels shift along the 3 axes giving a total of 125 search locations for each individual vector update.

$$\Phi^{n+1}_{ijk} = \underset{\phi \in S^n_{ijk}}{\text{minarg}} \; \Delta\hat{C}_\phi \tag{3.6}$$

$$S^n_{ijk} = \left\{ \phi \in \mathcal{Z}^3 \middle| \; |\phi - \Phi^n_{ijk}|_\infty \leq 2 \right\} \tag{3.7}$$

A simple estimate of the change of the probability distribution $p(a,b)$ caused by replacing Φ^n_{ijk} with ϕ can be obtained by assuming that the box of voxels associated with the vector Φ^n_{ijk} has a constant change of displacement. Let Q_{ijk} represent the set of coordinates indexing the template box of voxels associated with vector Φ_{ijk} so we can write

$$\begin{aligned}\Delta p_\phi(a,b) \simeq &\frac{1}{N} \sum_{\boldsymbol{v}\in Q_{ijk}} \delta\{a - \mathcal{A}(\boldsymbol{v}), b - \mathcal{B}(\boldsymbol{v}+\phi)\} \\ -&\frac{1}{N} \sum_{\boldsymbol{v}\in Q_{ijk}} \delta\{a - \mathcal{A}(\boldsymbol{v}), b - \mathcal{B}(\boldsymbol{v}+\Phi^n_{ijk})\}\end{aligned}$$

Using this result we get

$$\begin{aligned}\Delta D_\phi &\simeq \sum_{a,b} \Delta p_\phi(a,b) \frac{dD(\Phi^n)}{dp(a,b)} \\ &\simeq \sum_{\boldsymbol{v}\in Q_{ijk}} \frac{1}{N} \left\{ \frac{dD(\Phi^n)}{dp(\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v}+\phi))} - \frac{dD(\Phi^n)}{dp(\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v}+\Phi^n_{ijk}))} \right\} \\ &= c + \frac{1}{N} \sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{T}\{\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v}+\phi)\} \qquad (3.8)\end{aligned}$$

$$\mathcal{T}\{a,b\} = \frac{d}{dp(a,b)} D^n_\Phi \tag{3.9}$$

where c is independent of Φ_{ijk}^{n+1}. Equation (3.9) first applies a first order approximation[2] to the change in ΔD_ϕ wrt. $p(a,b)$, and secondly the aforementioned approximation to the change $\Delta p(a,b)$. Thus, the final change in D is estimated by summing values from a look-up table $\mathcal{T}(a,b)$ which can be computed from the current estimate of the $p(a,b)$ distribution, and has a size proportional to the square of the number of discrete levels that a and b may take, i.e. 256×256 if 8 bit internal representation is used. This look-up strategy makes the optimization of the global similarity measure computationally feasible since the bulk of computational work is reduced to indexing and summation operations. An update of an individual vector consists of, among all $\phi \in S_{ijk}$, choosing the one that minimizes the cost function, i.e. minimizing

$$\begin{aligned} \Delta\hat{C}_\phi &= \frac{1}{N}\textstyle\sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{T}\{\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v}+\phi)\} \\ &+ \alpha\textstyle\sum_{i'k'j'\in\mathcal{N}_{ijk}} \left\|\phi - \Phi^n_{i'j'k'}\right\|^2 . \end{aligned} \tag{3.10}$$

The vectors are updated in a 3D checker-board fashion, i.e vectors Φ_{ijk} for which $i+j+k$ is even are updated in the first round and the remaining vectors in the second round. This is done to increase the convergence speed of the cost function (3.1). Equation (3.10) shows that a vector update has a Markov property that each update depends only on the 6 nearest vector neighbors—this means that by updating first one half of the 3D checker-board and then the other half, the second half of updates will be experiencing the neighbors Φ^{n+1} field rather than the Φ^n field. This will in effect double the convergence rate.

Finally, the displacement fields are smoothed in between iterations by applying a 3D Gaussian spatial low-pass filter. The size of the Gaussian kernel FWHM, which we will denote β, is a smoothness parameter that needs to be carefully set together with the α of Equation (3.2).

Similarity Measures

Three different similarity measuring functions are described here, least squares (LS), correlation coefficient and the information based mutual information (MI). The look-up table $\mathcal{T}(a,b)$ is computed at each iteration step as listed in Table 3.2 . In case of the correlation and MI measures, $p(a,b)$ must be estimated between iterations, which involves the overhead of computing $\mathcal{B}(\boldsymbol{v}+\Phi^n_{\boldsymbol{v}})$. For the LS measure, $\mathcal{T}(a,b)$ may be computed on line, i.e. with no table actually computed which makes the LS method faster.

Method	Distance measure	$\mathcal{T}(a,b)$
Least Squares	$\sum_{a,b} p(a,b)(a-b)^2$	$(a-b)^2$
Correlation Coefficient	$\rho_{ab} = \frac{\sqrt{\sum_a p(a)(a-\bar{a})^2 \sum_b p(b)(b-\bar{b})^2}}{\sum_{a,b} p(a,b)(a-\bar{a})(b-\bar{b})}$	$(c_1(a-\bar{a}) + c_2(b-\bar{b}))^2$
Mutual Information	$-\sum_{a,b} p(a,b)\log\frac{p(a,b)}{p(a)p(b)}$	$-\log\frac{p(a,b)}{p(a)p(b)}$

Table 3.2: Lookup-table functions $\mathcal{T}(a,b)$ derived from 3 different similarity measures D, as described in Section 3.6. Note that constant terms and scaling factors are ignored in the $\mathcal{T}(a,b)$'s. Also note that the D listed for MI is actually the negative mutual information—the warp algorithm minimizes D, thereby *maximizing* the mutual information for $p(a,b)$. (c_1, c_2) is the eigenvector with smallest eigenvalue of the 2×2 covariance matrix between intensities a and b.

The LS using the equation in Table 3.2 is quite sensitive to global intensity variations in the MR scans. The LS algorithm referred to in the second part of the paper is actually a modified LS: The estimated change in the distance measure by replacing Φ^n_{ijk} with ϕ can be obtained in line with the approach used in Equation (3.9) using

$$\Delta D_\phi \simeq \sum_{\boldsymbol{v}\in Q_{ijk}} \frac{1}{N}\left\{\mathcal{A}(\boldsymbol{v}) - \bar{\mathcal{A}} - \mathcal{B}(\boldsymbol{v}+\phi) + \bar{\mathcal{B}}\right\}^2 \tag{3.11}$$

where $\bar{\mathcal{A}} = \sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{A}(\boldsymbol{v})/N_Q$ and $\bar{\mathcal{B}} = \sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{B}(\boldsymbol{v}+\Phi^n_{ijk})/N_Q$. This makes the algorithm fit the edges rather than the local intensity level of the data.

[2] It should be noted that the first order approximation poses a problem if D is not differentiable at $p(a,b) = 0$. We shall address this problem in Section 3.2.1 .

The estimation of $p(a,b)$ using (3.5) poses a problem in the case of MI. It is evident from Table 3.2 that $\mathcal{T}(a,b)$ breaks down in the neighborhood of $p(a,b)=0$. Increasing the bin size of the histogram $h^n(a,b)$ so that no bin gets a zero count would solve the problem but throws away intensity resolution. Increasing the bin-size can be thought of as smoothing the histogram with a box-filter, so instead of using the same filter size all over the histogram, an adaptive smoothing is performed. Areas of $h^n(a,b)$ with low counts are smoothed with a large scale kernel (similar to a large bin), areas with higher counts with a smaller kernel. The kernel used is a Gaussian kernel with a FWHM adjusted so that the signal to noise ratio of the smoothed histogram counts is approximately constant, assuming local uniformity of $p(a,b)$. The signal to noise ratio is computed from the variance in the binomial distribution: Let $h(a,b)$ be the current histogram of matched intensities and g_f a Gaussian blurring kernel of FWHM equal to f. Assuming local uniformity of $h(a,b)$ we may estimate a local signal to noise ratio (SNR) using

$$\begin{aligned} \mathrm{SNR} &\simeq \frac{h(a,b)}{\sqrt{\mathrm{Var}\{h * g_f(a,b)\}}} \\ &\simeq \frac{h(a,b)}{\sqrt{\sum_{a',b'} g_f^2(a',b')h(a-a',b-b')}} \\ &\simeq \sqrt{\frac{\pi h(a,b) f^2}{2\log 2}} \end{aligned} \tag{3.12}$$

so that

$$f \simeq \mathrm{SNR}\cdot\sqrt{\frac{2\log 2}{\pi h(a,b)}} \tag{3.13}$$

In the actual implementation the histogram is smoothed with 5 blurring kernels (FWHM = 2, 3, 5, 7 and 10 histogram indices). For each (a,b), Equation (3.13) selects the correct local blurring size among these 5 histograms. In order to stabilize the procedure, a blurred histogram (FWHM=2) is used for $h(a,b)$ in (3.13). A SNR value of 10.0 produced good results. After applying this scheme, no bins will contain the zero count value (because of the global range of the Gaussian kernel), and the parts of the histogram that have a high number of counts preserve their information.

Discussion

Both the sum of squares and the correlation similarity measures rely on the fundamental assumption that the residual difference between the warped subject image and the template is white Gaussian noise. This can be realized by thinking of the cost function (3.1) as a negative log posterior distribution. The regularizer R is a prior distribution on the displacement vectors while the similarity measuring term D is a model of the residual noise, which in the quadratic cases is a product of independent Gaussian distributions. We know, however, that this assumption is incorrect in the sense that the subject anatomy *is not* a deformed instance of the template anatomy—real anatomical differences certainly exist among normal brains. The major disadvantage of the squared similarity measures is that the anatomical differences that cannot be corrected for by the warp are penalized with a factor depending on how the image modality represents the mismatched tissue types as image intensities. There is no reason to believe that a mis-registration of e.g. gray matter with CSF should be penalized by a factor depending only on the difference in the intensity representations of those two tissue types. The MI on the other hand, penalizes (mis)matches with a factor depending on the resulting specificity (i.e. entropy of the conditional distribution $P(b|a)$ of subject voxel intensities b given corresponding template voxel intensity a) of the mapping between image intensities. This is certainly a more general approach, and it may be a better criteria.

Another relevant issue is the number of iterations needed for the warp to converge at each resolution step. Our experience has been that MI requires twice the number of iterations to converge (typically 8) compared to the sum of squares method (typically 4) so that the total computation time is approximately doubled. This is because the probability distribution $p(a,b)$ is a more complex function of the displacement field when using MI than LS or correlation, so that more iterations are needed in each resolution step.

Smoothness Constraints

Several factors are working together to ensure the smoothness of the displacement field. The reason for requiring smoothness is simply a question of controlling noise in the resulting displacement map. The factors are:

- *Use of a regularizer, or prior term in the global cost function (3.1).* The term $R(\Phi)$ is essentially a Gaussian Markov field prior. The frequency power spectrum for this prior may be calculated analytically, see e.g. [52]. Previous work has been published applying physical/mechanical prior distributions to the deformation problem [42, 53, 54, 55, 56, 57]. However, considering that the deformation itself is non-physical (i.e. the subject anatomy *is not* a deformed instance of the template anatomy), we argue that neither does the prior distribution have to be.

- *Controlling the sampling of the displacement field.* By limiting the number of vectors in the representation of Φ we efficiently limit the upper frequency components of the deformations that can be represented. In [58], the displacement field is expressed as an expansion of smooth basis functions thereby restricting the resolution of the resulting displacement field. In [46] a hierarchical search for the displacement field is performed by representing the displacement field in wavelet components, and gradually increasing the number of components used.

- *Smoothing of the image volumes.* This technique is used implicitly in the sub-sampling of the subject and template volumes. It is also used in [58] and in the ANIMAL algorithm described in [59, 39, 40, 43].

- *Direct smoothing of the vector field components.* This approach is a direct way of shaping the frequency characteristics of the displacement field. The parameter β controls the size of the Gaussian low-pass filter used. Low-pass filtering of the displacement field vectors is also used in [38].

The problem of actually choosing the parameters used is a topic of current investigation. At each resolution step a new set of parameters is applied. Preliminary work based on the effect of the warp on the analysis of a functional data set has indicated that α and β cannot substitute for one another, although there exists a correlation in the optimal (with respect to the resulting functional alignment as described in Section 3.4) setting of the two parameters. It should further be noted that the setting of α depends strongly on the voxel similarity measure used, since the effect of the regularizing term changes with the dynamics of the match term in (3.1).

3.3 Data Acquisition

The functional activation study was a saccadic eye movement experiment [60] done at Rigshospitalet, Copenhagen, Denmark, consisting of 5 male right-handed subjects, each scanned 8 times (2 baseline, 6 graduated target frequency) during two scanning sessions, over two days with randomized ordering of scans. An array of LED's was located in an arc at angles -40° to 40°. The subjects performed visually guided voluntary anti-saccades. This paradigm involved subjects fixating on a central LED until a random target LED was lit. The task was to perform a saccade to the LED contra-lateral to the lit target. The target presentation frequency was varied from scan to scan from 0.05, 0.1, 0.3, 0.5, 0.7 to 0.9 Hz. Two baseline scans were acquired with the subject fixating at the central LED. The scanner is a whole body PET-scanner (Advance, General Electric, [61]) with an image resolution of approximately 5 mm in all directions. The scans were reconstructed with 35 slices, voxel size $2.0 \times 2.0 \times 4.25$ mm covering a field of view of $25.6 \times 25.6 \times 15.2$ cm. In addition, MR scans were obtained for the 5 PET subjects plus an additional subject to act as an alignment template. The scans were T1-weighted acquisitions using a Siemens Magnetom Vision scanner located at Hvidovre Hospital, Copenhagen, with $0.98 \times 0.98 \times 1.17$ mm voxel size for two of the five subjects and $0.98 \times 0.98 \times 1.00$ mm voxel size for the remaining 4 subjects, all acquired using a 3D MPRAGE sequence. The PET experiment consisted originally of 8 subjects out of which 2 did not wish to have the MR scan, and 1 subject was removed because his MRI showed an abnormality in the cerebellum, leaving 5 subjects for this experiment.

3.3.1 Preprocessing

A common preprocessing step was performed on all scans. Each subject's PET scans were intra-subject aligned using a 6 parameter rigid body rotation and translation (AIR, [6]) to match each scan to the subject's first scan. A mean scan was computed for each subject and a slice based threshold of 45% of the slice maximum was applied. The injected dose/weight ratios were not available, so global scaling of each scan was achieved using a voxel based Scaled Subprofile Model (SSM) analysis ([62, 63], see Section 3.4). No spatial smoothing was applied to the scans following the reconstruction, yielding an effective resolution of approximately 5 mm.

The MR scans were corrected for intensity inhomogeneity, which was predominant in the axial (Z) direction. This was accomplished by computing the mean of the top 5% intensity voxels for each slice containing brain tissue, and normalizing the slice intensities by fitting a 6'th order polynomial through these values as a function of the slice number, which gave a visually improved result. The MR scans were then carefully stripped from skull and scalp using a manual interactive stripping/region drawing tool, retaining mainly CSF, white matter and grey matter within a contiguous mask.

3.3.2 Three Alignment Schemes

Three registration schemes were set up for comparison as illustrated in Figure 3.2. Scheme A was our standard alignment procedure using a 12-parameter affine registration to align the mean PET images for each subject with a Talairach PET template volume obtained by simulating a PET scan from a high-resolution segmented MRI scan [33]. In Scheme B the 12-parameter inter-subject registration was computed using the subject's stripped MR volumes and a Talairach transformed MR template brain, and the PET images were aligned via this transformation, so that each PET scan was registered by combining the individual 6 parameter PET-PET and PET-MR transformations, and 12-parameter affine transformations in a single re-sampling operation. In Scheme C–E a 3D non-linear registration field was computed on each of the subject's 12-parameter registered, stripped MR scans to match the stripped template scan. Each PET scan was resampled combining the non-linear warp, the individual 6 parameter PET-to-PET, PET-to-MR, and 12-parameter MR to template transformations in a single re-sampling operation.

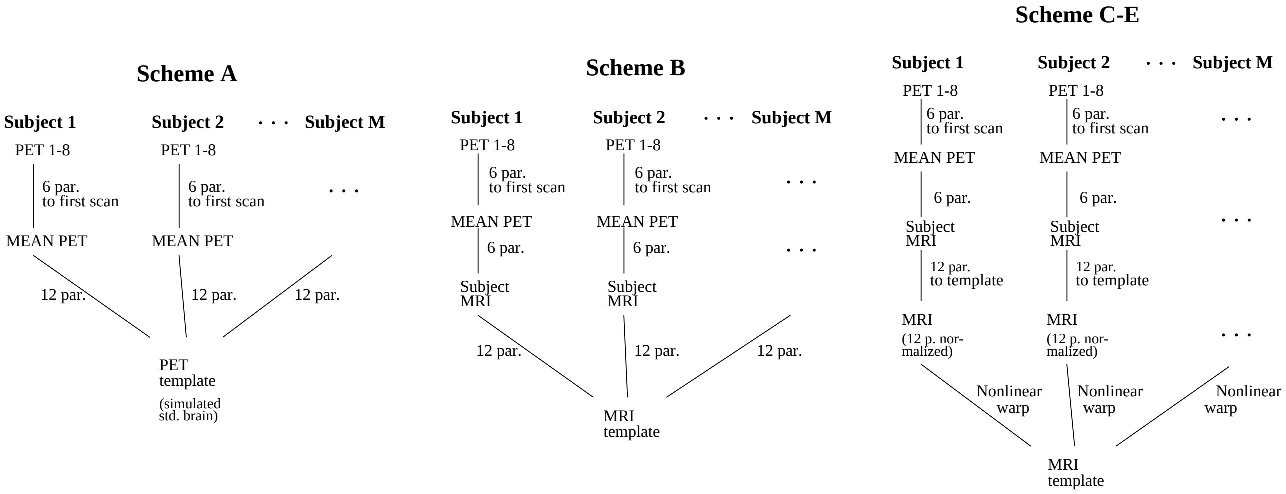


Figure 3.2: The three alignment schemes used for co-registering PET images across subjects.

In all three cases, both the raw PET images and the binary masks (1 is tissue, 0 is background) generated from the mean PET images were resampled using tri-linear interpolation in the same resolution as the original data ($2 \times 2 \times 4.25$ mm). New binary subject masks were formed by thresholding the resampled masks at 0.5. All 5 resampled masks were then logically AND'ed to find the resulting global mask used for the functional analysis.

Apart from the absence of a step with spatial smoothing of the PET images, Scheme A was identical to our standard procedure used for group analysis of PET activation studies also described in [64, 33]. This is a relevant comparison since in many cases, MRI scans are not acquired routinely with the PET scans.

One may argue that method B and C are more sensitive to errors in the 6 parameter registration than method A since they concatenate two rigid body transformations while method A only involves one. This suggests the alternative procedure of aligning each of the subject's PET scans to the MR scan individually. That approach has not been used because it is our experience that the noise in the raw PET scans gives problems with the cross modality registration, so that the total registration error becomes larger. A possible solution to this dilemma for future investigation is offered in [34] by reconciling all possible pairwise rigid body transformations using an iterative optimization scheme.

For comparison, three similarity measures for the non-linear warp were selected: The modified least square (LS), correlation coefficient (CC) and mutual information (MI), producing three separate non-linearly aligned data sets (which we shall refer to as C, D and E, respectively). At this point we face the problem of setting the regularization parameters α Equation (3.10) and β of Section 3.2.1 . Three resolution steps were used ($64 \times 64 \times 40$, $128 \times 128 \times 80$, and $256 \times 256 \times 160$) and the regularization parameter α was set to 150, 35 and 6 for the three resolution steps for the warp using LS as well as CC, and to 0.275, 0.066 and 0.011 for the warp using MI. These values were chosen so that the average variation of the regularization term in Equation (3.10) achieves the same order of magnitude as the average variation of the similarity term during the warp iterations. The filtering parameter β which controls the Gaussian FWHM (measured in vector indices) of the direct smoothing of the vector field between iterations chosen to 0.65, 0.41 and 0.38 for the three resolution steps, or 2.60, 1.64 and 1.52 image voxels, respectively.

3.4 Data Analysis

3.4.1 Multivariate Functional Analysis

Schemes A, B and the three different non-linear registrations for Scheme C produced five sets of stereotactically registered PET scans. These scans of each data set were passed through an SSM analysis [62, 63] and a subsequent Canonical Variate Analysis (CVA, see e.g. [65, Ch. 12]) of the resulting eigenvectors.

The SSM model is based on an orthogonal decomposition of the covariance structure of the log-transformed data, with a global scaling factor g_q for each scan. Let v_{iq} denote the detected $\left[{}^{15}\mathrm{O}\right]$water concentration at each voxel location i in scan q. With this notation, the SSM model is

$$
\begin{aligned}
v_{iq} &= g_q\left(i_{iq} + r_i\right), \qquad \sum_i i_{iq} = 0 && (3.14)\\
i_{iq} &= \sum_k^K h_{ik}\sqrt{\lambda_k}s_{kq} = \sum_k^K h_{ik}\mathrm{SSF}_{qk}, && (3.15)
\end{aligned}
$$

where K is the number of orthogonal components in the model. In the analysis used in the following, K equals the estimated number of degrees of freedom in the data matrix. The vector r_i is the mean activation image, and the i_{iq} matrix contains the interaction terms. This interaction matrix is decomposed using the orthonormal vectors h_k each of which describe a covariance relationship between the voxels. The eigenvectors SSF_{qk} have corresponding eigenvalues λ_k, $\lambda_1 > \lambda_2 > \ldots > \lambda_K$, which describe the amount of variance accounted for by the k'th eigen-vector[3].

The SSM analysis is purely data-driven, which means that no information about the particular activation (paradigm) state goes into the model. Nevertheless, in multi-subject activation experiments it is commonly the case that the first M-1 eigenvectors (M equals the number of subjects) describe variation between individuals and the following eigenvectors describe variation related to the activation experiment, see e.g. [66, 67]. This division of signal variance into an inter- and an intra-subject component is also visible in Figure 3.6from the drop in the magnitude between eigenvalues 4 and 5 and was also found in the original group of 8 subject's PET scans [68].

For the particular experiment there were 6 different saccadic frequencies and 1 fixation condition. The variation between these 7 groups may be analyzed within the space spanned by the SSM eigenvectors using a Canonical Variate Analysis (CVA) [65, Ch. 12] to find a low dimensional subspace containing all the experimental structure as previously utilized in [67, 69]. The CVA analysis generates an orthogonal decomposition of the between group variance, so that the n'th canonical vector holds the n'th largest ratio of between group covariance and within group covariance. The CVA is performed only in the subspace spanned by the 12 largest SSF eigenvectors. The eigenvalues of the CVA analysis describe the distribution of the total variance into the respective canonical vectors. The computational details are outlined the Section 3.5.

[3] If $i_{iq} = \sum_{k=1}^{K} \mathrm{SSF}_{qk} h_{ik}$ then the matrices h_{ik} and SSF_{qk} are in essence the components of a Singular Value Decomposition (SVD, see e.g. [8] although they are not computed directly as such because of the g_q factors, see [62]) of the interaction matrix i_{iq}, in matrix notation: $i_{iq} = (\mathbf{USV}^T)_{iq}$ with $\mathbf{U}_{ik} = h_{ik}$ and $(\mathbf{SV}^T)_{qk} = \mathrm{SSF}_{qk}$.

Univariate Variances

More direct univariate variance measures were also computed. Let scan index $q = 1 \dots Q$ be divided into subject $m = 1 \dots M$ and session scan number $l = 1 \dots L$. The total inter- and intra-subject variances normalized by the number of voxels retained, I, are computed from the interaction i_{iq}:

$$\begin{aligned}
\sigma^2_{\text{TOTAL}} &= \frac{1}{I(Q-1)} \sum_{i,l,m} (i_{ilm} - i_{i\cdot\cdot})^2 = \frac{1}{I(Q-1)} \sum_{i,l,m} i^2_{ilm} \\
\sigma^2_{\text{INTRA}} &= \frac{1}{I(Q-M)} \sum_{i,l,m} (i_{ilm} - i_{i\cdot m})^2 \\
\sigma^2_{\text{INTER}} &= \frac{1}{N(M-1)} \sum_{i,m} (i_{i\cdot m} - i_{i\cdot\cdot})^2 = \frac{1}{I(M-1)} \sum_{i,m} i^2_{i\cdot m}
\end{aligned} \tag{3.16}$$

3.4.2 Anatomical Analysis

The inter-subject variance may also be computed from the MR images. Figure 3.3shows selected orthogonal slices of the template MR and the averaged registered MR scans for each of the results of 12-parameter alignment, and warp using MI and LS. The left column is the template volume, and the three other columns are the averaged aligned MR volumes using the three methods. Clearly, the warp using MI as similarity measure produces images that vary the least from the template volume, so that the resulting mean image is sharpest, followed by the warp using CC and LS.

The non-linear registrations resulted in a better alignment of the gross shape of the brain. The MI seems to have performed a slightly better alignment than CC and LS, but none of the registration methods were able to perfectly match the cerebral sulci.

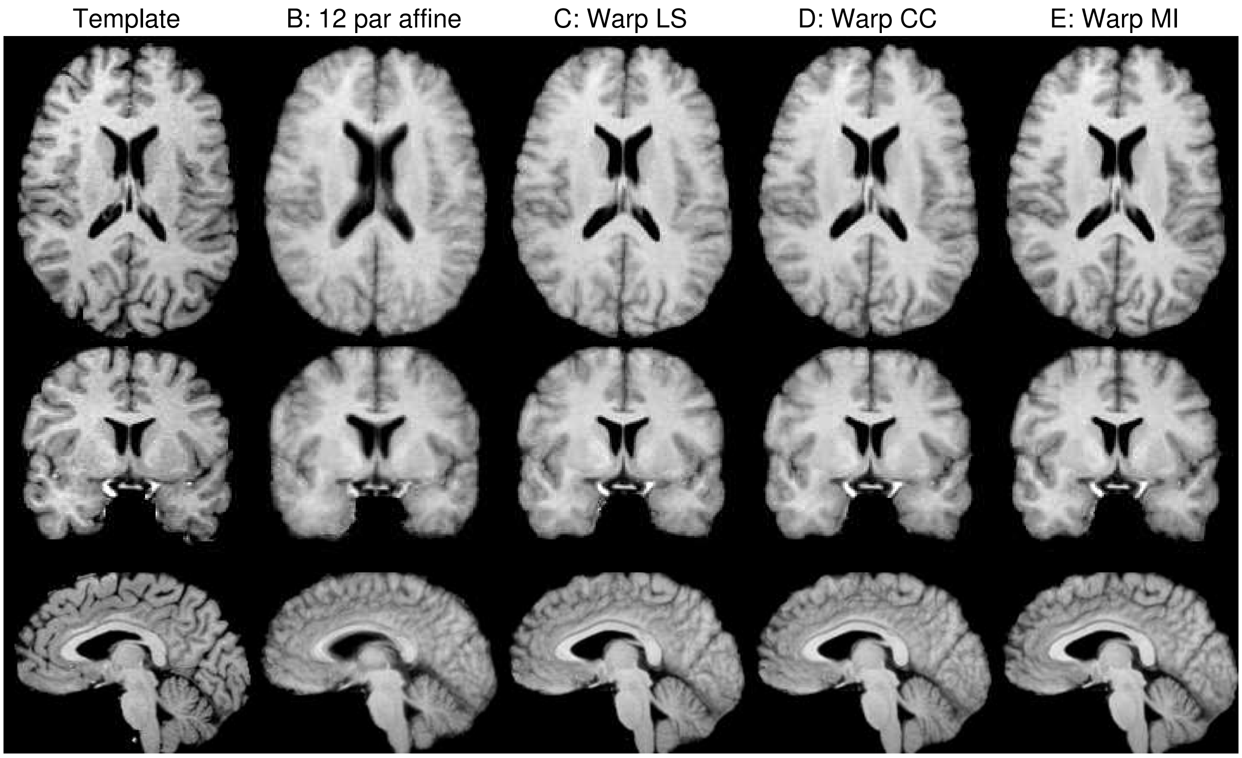


Figure 3.3: Comparison of MR alignments. Leftmost image is a horizontal slice of the template volume. The four other images are mean slices across aligned subjects of the three alignment methods : 12-parameter affine and warp using the LS, CC and MI similarity measures respectively. The sharper image of MI indicates better alignment by this method.

Method	A	B	C		
	12p PET-PET	12p via MRI	C: Warp/LS	D: Warp/Corr	E: Warp/MI
[a]MRI σ_{INTER}	-	30.5	26.6	25.4	24.8
N.o. PET voxels retained, I	67621[b]	81689	83349	82908	83773
σ^2_{TOTAL}	$2.089{\cdot}10^{-2}$	$2.060{\cdot}10^{-2}$	$1.832{\cdot}10^{-2}$	$1.764{\cdot}10^{-2}$	$1.752{\cdot}10^{-2}$
σ^2_{INTER}	$1.349{\cdot}10^{-2}$	$1.315{\cdot}10^{-2}$	$1.024{\cdot}10^{-2}$	$0.946{\cdot}10^{-2}$	$0.913{\cdot}10^{-2}$
σ^2_{INTRA}	$1.094{\cdot}10^{-2}$	$1.093{\cdot}10^{-2}$	$1.105{\cdot}10^{-2}$	$1.100{\cdot}10^{-2}$	$1.117{\cdot}10^{-2}$
SSM: $\sum_{i=1}^{4} \lambda_i/N$	$1.108{\cdot}10^{-2}$	$1.082{\cdot}10^{-2}$	$0.844{\cdot}10^{-2}$	$0.780{\cdot}10^{-2}$	$0.753{\cdot}10^{-2}$
CVA: $\lambda_1/\sum_{i=2}^{5} \lambda_i$	5.505	5.803	7.653	7.397	7.307

[a]MRI σ_{INTER} is measured per voxel inside brain mask.
[b]The number of voxels retained in the PET-PET analysis (A) is not comparable to the MRI registrations because the PET template was smaller than the MRI template used in Schemes B and C–E.

Table 3.3: Comparison of effect of the warp vs. affine on selected variance measures per voxel from the anatomical (MRI) and functional (PET) analysis.

3.4.3 Functional Analysis Results

Numerical results of the analysis are summarized in Table 3.3 . Note that the non-linear registration retains more voxels in the analysis, the warp using MI retained 2.5% more voxels than the 12-parameter method. This is occurs because the non-linear registration fits the gross shape of the brain quite well. We only retain voxels that are present in all scans, and this makes the analysis quite sensitive to registration errors at the edges of the masks—in this way the number of voxels becomes a measure of registration performance, and the non-linear registration is seen to best preserve the field of view of the analysis. Since most of the extra voxels in the non-linear registration analysis reside in grey matter areas of the cortex, we may be retaining important information about cortical activation that would be excluded using the 12-parameter alignment. Note that the number of voxels for Scheme A (PET-to-PET registration) is not directly comparable to the other schemes since the PET template brain was smaller than the MR template used in Schemes B and C–E. Figure 3.4shows horizontal slices (+45 mm above and 10 mm below AC-PC plane) of the first canonical vector, i.e. the direction in scan space of largest functional discrimination between the groups representing saccadic frequencies. It is evident that the non-linear registration (the two right-most images) is better at preserving edge voxels in the analysis. The upper slice contains an activation focus in the dorso-lateral pre-frontal cortex, which was undetected by the 12-parameter affine registration scheme B because the voxels near the edge of the brain were masked out. The major activation foci in the visual cortex were captured by all the registration schemes tested.

The change in total, inter- and intra-subject variance is shown graphically in Figure 3.5: The drop in total variance ($2.09/2.06 \cdot 10^{-2}$ affine A/B to $1.85/1.76/1.75 \cdot 10^{-2}$ non-linear C/D/E) is seen to arise mainly from a change in the inter-subject variance; the intra-subject variance is virtually unaffected by the alignment scheme. The inter-subject variance drop is absorbed mainly by the first 4 eigenvectors of the SSM analysis, as can be seen in Figure 3.6. The plot shows the SSM eigen-spectra (λ_k) with each eigenvalue normalized by the number of voxels retained in the analysis. Also, the sum of the first 4 eigenvalues listed in Table 3.3 reflects the change in inter-subject variance σ^2_{INTER}.

These results indicate that Scheme E (warp using MI) may be the superior registration method. But until now, we have been looking only at very direct variance measures, and we have seen that Scheme E has the largest impact on the functional alignment and reduces the inter-subject variance the most. This way of evaluating the registration does not measure how good a *functional* alignment we have achieved, i.e. how good is the signal to noise ratio of the observed brain signal. In the following we shall investigate how well the brain state is reflected in the scans, assuming that the optimal alignment is the one that is able to extract the brain state most clearly. A ideal approach would be to measure the reproducibility or generalization

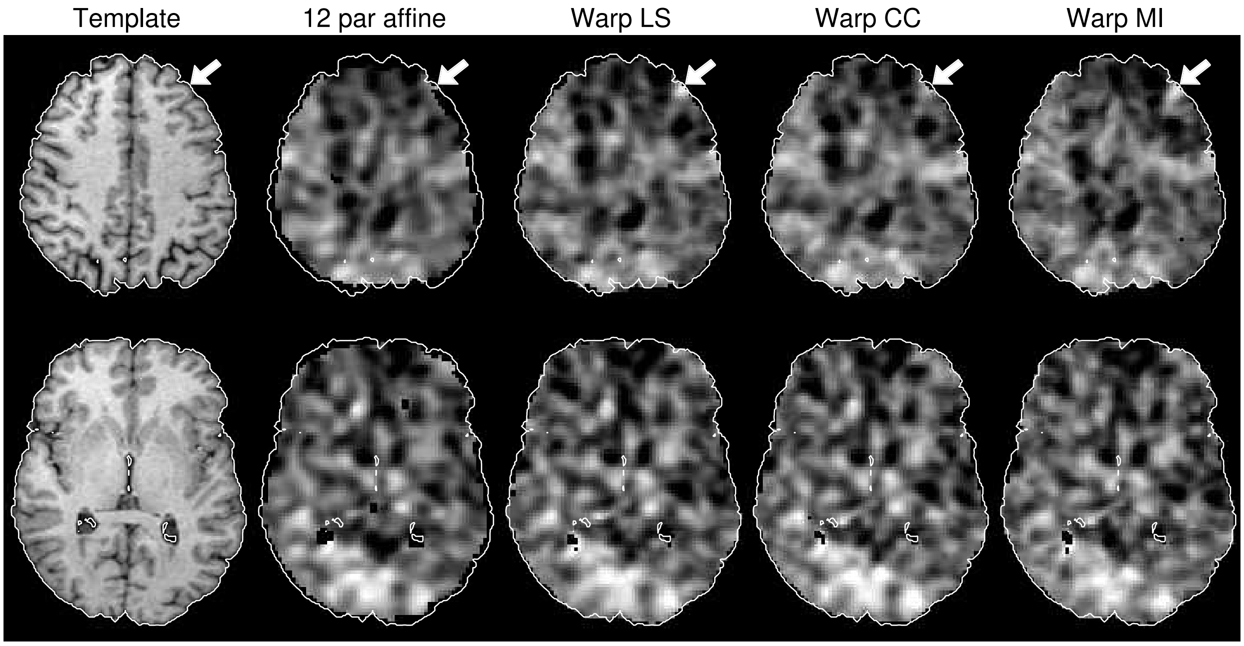


Figure 3.4: Comparison of functional activation patters. Leftmost image columns shows horizontal slices (+45 mm above AC-PC plane and −10 mm below) of the template volume. The four other columns shows corresponding slices of first canonical component as obtained from the analysis of aligned subjects of the four alignment methods : 12-parameter affine and warp using the LS, CC and MI similarity measures respectively. Note that the increased number of voxels in the non-linear cases results in a larger field of view of the end analysis, causing the hot spot in the dorso-lateral prefrontal cortex to be overlooked by the affine methods.

ability of the spatial activation patterns obtained, but this would require a data set with many more than 5 subjects as investigated here (e.g. [70]).

The canonical variate analysis gives information about how the frequency of the saccadic eye movement affects the scans in a low-dimensional subspace embedded in a high-dimensional space. All scans are grouped according to the saccadic eye movement frequency (with 7 groups in all). The canonical vectors form an orthogonal basis that separates these groups the most with respect to the spread within groups. If the relation between the scan and the saccadic frequency is purely linear, the variation between the 7 frequencies can be described adequately with a linear model. Figure 3.7shows a plot of the canonical projections for all 40 scans registered using Scheme B. The circles indicate 90% confidence areas for the location of each saccadic frequency group mean [65, Ch. 12]. Although mild nonlinearities in the saccadic functional activation patterns from this experiment have been reported [68, 71], Figure 3.7demonstrates that the majority of the group structure is well described by the first canonical component. We therefore assume that the remaining canonical components (λ_2–λ_6) describe 'noise' in the activation, i.e. patterns not directly related to the paradigm. Thus, the ratio of the first canonical component to the sum of the rest ("Signal to Noise Ratio") is an indication of how well the brain activation signal is captured. As shown in Table 3.3 it is actually Scheme C with a ratio of 7.7 which performs best in this sense, while Scheme E with the lower ratio of 7.3 produced the lowest inter-subject variance. We also see that the non-linear registration methods tested all outperform the 12 parameter affine registration of Schemes A and B. Figure 3.8shows a scatter plot of the first versus second canonical eigenvalue from the CVA analysis for each of the 5 alignment Schemes A–E.

These results seem to indicate that it is possible to 'over-warp' the data: Instead of registering functionally equivalent areas among subjects, the more detailed registration (e.g. Scheme E) applies more active deformations which *degrade* the functional registration. To illustrate this, we compute the Jacobian of the non-linear transformations. Histograms of Jacobian values (excluding background voxels) averaged over the 5 subjects are shown in Figure 3.9. The transformation computed by MI involves more compression ($J < 1$) and expansion ($J > 1$) of voxel elements than the transformation generated by CC and LS.

The above discussion thus leads us to conclude that although the warp of Scheme E does the best job at reducing inter-subject variation, we seem to be better off applying a registration method which does not go

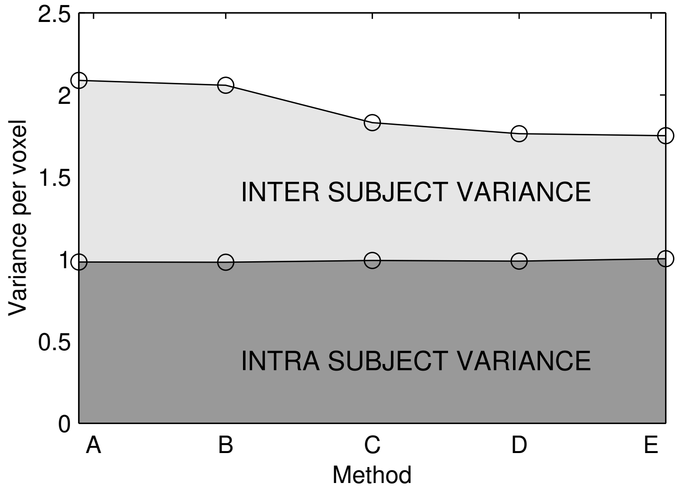


Figure 3.5: Division of the total functional variance into inter- and intra subject contributions. A: Alignment using PET to PET 12-parameter affine; B: 12-parameter affine via MRI; C–E: Non-linear warp using a LS, CC and MI respectively as similarity measures.

into the same level of detail. Presumably this is an issue of choice of regularization parameters for the warp, rather than an issue of selecting among similarity measures, but it is our experience that the MI similarity measure offers the possibility of performing registration at a higher level of detail than is accomplished with the CC and LS measures.

3.5 Canonical Variate Analysis

The canonical variate analysis stems from the multivariate analysis of variance and may be thought of as the analogue of PCA for grouped data. Consider an experiment with J conditions (or groups) and n_j, $j = 1 \ldots J$ samples of the distribution

$$\boldsymbol{x}_{ij} \in N(\boldsymbol{\mu}_j, \Sigma), \qquad i = 1 \ldots n_j,\ j = 1 \ldots J$$

so that each $\boldsymbol{x}_{ij}$ is a vector of size $p \times 1$. One may test if the $\boldsymbol{\mu}_j$'s are different, i.e. whether the hypothesis $H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \ldots = \boldsymbol{\mu}_J$ can be rejected. This is done using a standard likelihood ratio test, see e.g. [65, Ch. 12]. If we are using the CVA to analyze functional brain activation in the saccadic eye movement experiment, we can group the scans according to the paradigm, in this case the saccadic frequency. Each vector $\boldsymbol{x}_{ij}$ represents[4] a brain scan with LED presentation frequency indexed by j and subject and repetition index i. Usually H_0 will be rejected (unless the paradigm did not induce significant brain activation) so we may ask more specifically about the dimensionality of the space spanned by the $\boldsymbol{\mu}$'s. Note that this is different than the univariate ANOVA case where the dimensionality is either 0 or 1. The canonical analysis performs an orthogonal decomposition of this space, so that we can inspect and test the dimensionality. Let

$$\begin{aligned}
\mathbf{W} &= \sum_{i,j} (\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_j)^T, \\
\mathbf{T} &= \sum_{i,j} (\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}})(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}})^T, \text{ and} \\
\mathbf{B} &= \sum_{i,j} n_j (\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T
\end{aligned} \tag{3.17}$$

[4]Traditionally, the $\boldsymbol{x}_{ij}$ would represent an entire scan, but this causes problems in the canonical analysis since the within group covariance matrix $\mathbf{W}$ (which is a $p \times p$ matrix) becomes singular when $p > Q - J - 1$, where is the total n.o. scans. A way to circumvent this problem is to use the p largest SSM eigenvectors to represent the scans in a lower dimension. In other words, x_{ij} represents the p values of the i'th scan in the j'th group for the p SSM eigenvectors with the largest eigenvalues. The dimensionality p should be chosen so that i) it is large enough to embed the expected variation introduced by the paradigm ii) smaller than $Q - J - 1$. In the experiment of saccadic eye movements described in this paper, p was 12.

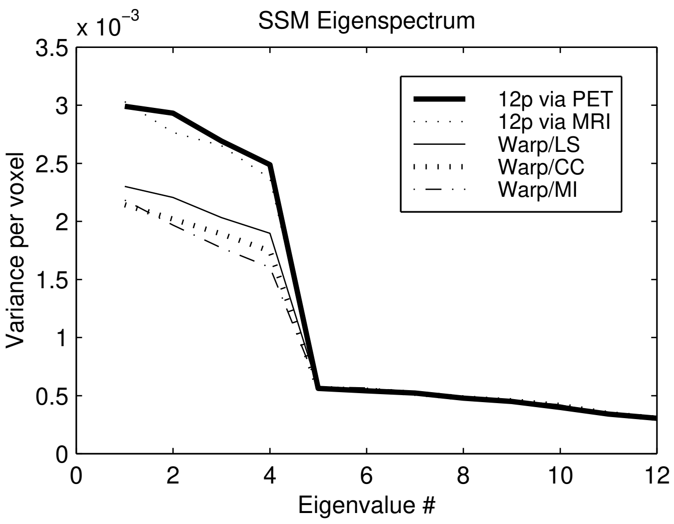


Figure 3.6: The plot shows the SSM eigenspectrum obtained from each of the 2 affine and 3 non-linear alignment experiments.

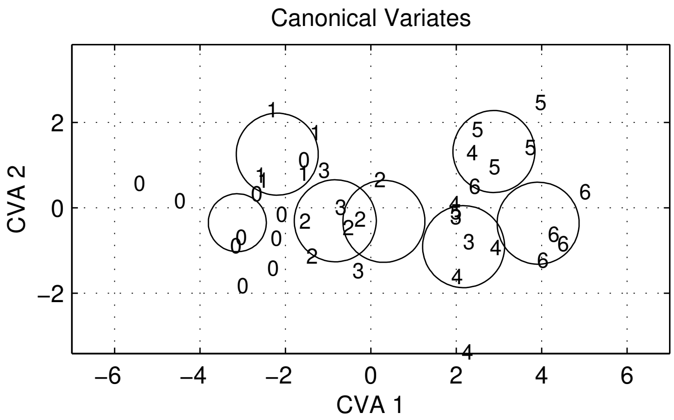


Figure 3.7: Plot of first versus second canonical projections for the CVA analysis of the groupings 0: fixation, 1–6: 0.05, 0.1, 0.3, 0.5, 0.7 and 0.9 Hz saccadic eye movements. The circles are rough 90% confidence areas for the group means. The analysis was run using the 12-parameter affine registration via MRI (Scheme B).

where $\bar{\boldsymbol{x}}_j$ is the mean of the group that scan number j belongs to and $\bar{\boldsymbol{x}}$ is the global mean, so that $\mathbf{T} = \mathbf{W} + \mathbf{B}$. The canonical analysis proceeds as the eigenvector analysis of the matrix $\mathbf{W}^{-1}\mathbf{B}$. The r'th canonical vector is the r'th eigenvector of this matrix and describes the linear function which obtains the largest separation of the J groups under the condition that the vector is orthogonal to the $r-1$ previous canonical directions. Note that the first canonical variate is equivalent to the Fisher's linear discriminant for optimal linear discrimination between groups. The CVA analysis can be understood as the principal components analysis of the between group covariance matrix $\mathbf{B}$ in a coordinate system transformed so that the within group covariance is unity. A scatter plot of the 1st versus 2nd canonical coordinates as shown in Figure 3.7can give a useful picture of the structure in the data. In such a plot, the axes are in units of standard deviations of a single data point, and we get an indication of the strength of the separation of the individual groups.

3.6 Derivation of the Look-up Table for Warp Similarity

This section describes the derivation of the voxel similarity look-up table functions listed in Table 3.2. The look-up table functions are used to estimate the change of the global similarity measure arising from a small

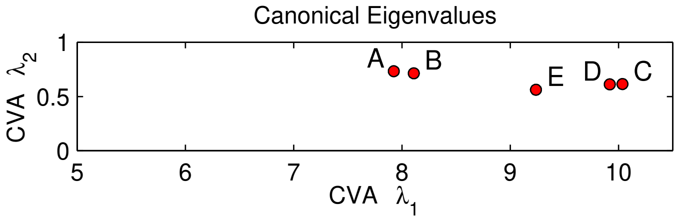


Figure 3.8: Scatter plot of first versus second canonical eigenvector of a canonical analysis of the distribution of 7 functional activation states. The non-linear registration methods (C, D& E) achieve a larger 1st eigenvalue indicating better alignment. A: Alignment using PET to PET 12-parameter affine; B: 12-parameter affine via MRI; C–E: Non-linear warp using a LS, CC and MI respectively as similarity measures.

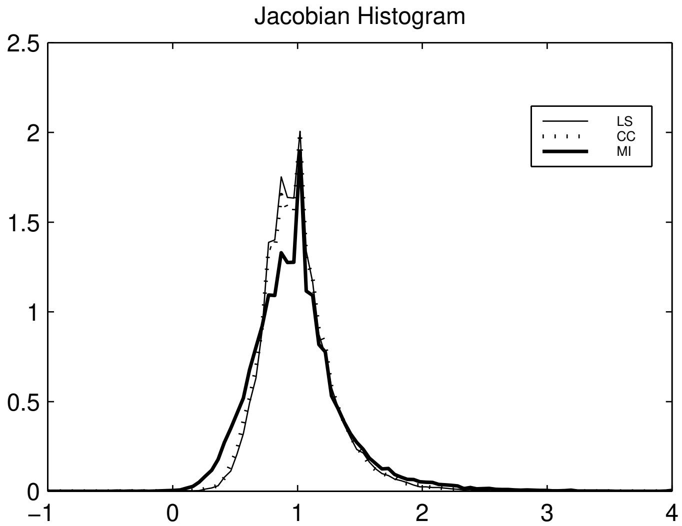


Figure 3.9: Histogram of Jacobian values for the transformations computed by MI and LS. Background voxels are left out. Notice that the MI transformation has more compression ($J < 1$) and expansion ($J > 1$) of voxels than both CC and LS indicating a more active deformation.

change in the histogram of matched voxel intensities. In the following it is assumed that the histogram is converted into a probability function by normalizing the discrete histogram $p(a, b) = h(a, b)/\Sigma_{a,b} h(a, b)$.

Mutual Information

Mutual information is defined by

$$D_{\mathrm{MI}}[p(a,b)] = \sum_{a,b} p(a,b) \log \frac{p(a,b)}{p(a)p(b)} \tag{3.18}$$

The lookup-table function is defined as $dD_{\text{MI}}[p(a,b)]/\,dp(a',b')$ with the constraint that $\sum_{a',b'} p(a',b') = 0$ which is found by straigh-forward algebra

$$\frac{d\mathrm{MI}[p(a,b)]}{dp(a',b')} = \sum_{a,b}\left[\delta(a-a',b-b')\log\frac{p(a,b)}{p(a)p(b)} + p(a,b)\frac{d}{dp(a',b')}\log\frac{p(a,b)}{p(a)p(b)}\right] \tag{3.19}$$
$$= \log\frac{p(a',b')}{p(a')p(b')} + \sum_{a,b} p(a,b)\left(\frac{\delta(a-a',b-b')}{p(a',b')} - \frac{\delta(a-a')}{p(a')} - \frac{\delta(b-b')}{p(b')}\right)$$
$$= \log\frac{p(a',b')}{p(a')p(b')} + \left(\frac{\sum_{a,b} p(a,b)\delta(a-a',b-b')}{p(a',b')} - \frac{\sum_{a,b} p(a,b)\delta(a-a')}{p(a')} - \frac{\sum_{a,b} p(a,b)\delta(b-b')}{p(b')}\right)$$
$$= \log\frac{p(a',b')}{p(a')p(b')} + \left(\frac{p(a',b')}{p(a',b')} - \frac{\sum_{a,b} p(a',b)}{p(a')} - \frac{\sum_{a,b} p(a,b')}{p(b')}\right)$$
$$= \log\frac{p(a',b')}{p(a')p(b')} + \left(1 - \frac{N_a p(a')}{p(a')} - \frac{N_b p(b')}{p(b')}\right) = \log\frac{p(a',b')}{p(a')p(b')} + \text{const.}$$

which means that a finite change $\Delta h(a,b)$ in the observed histogram results in the estimated change in the mutual information

$$\Delta D_{\text{MI}} \simeq \sum_{a,b} \Delta h(a,b)\log\frac{p(a,b)}{p(a)p(b)}. \tag{3.20}$$

Entropy

Entropy is an unstable voxel similarity measure since it prefers to map all intensities to the same level (e.g. background) rather than assigning individual intensities in the template to separate intensities in the subject volume. The derivation of the lookup-table function is simple

$$D_{\text{EN}}[p(a,b)] = \sum_{a,b} -p(a,b)\log p(a,b) \tag{3.21}$$

$$\frac{d\mathrm{EN}[p(a,b)]}{dp(a',b')} = -1 - \log p(a',b') \tag{3.22}$$

A finite change $\Delta h(a,b)$ in the observed histogram yield the estimated change of similarity

$$\Delta D_{\text{EN}} \simeq -\sum_{a,b} \Delta h(a,b)\log p(a,b). \tag{3.23}$$

Correlation Coefficient

Maximizing the correlation coefficient corresponds to gathering all samples in the histogram $h(a,b)$ along a single line. In stead of minimizing the correlation coefficient directly we will adopt a look-up table which behaves dynamically like the squared error similarity measures. This has the advantage that the warp parameter α gets approximately the same parameter value as with the squared error functions, which aids the problem of determining appropriate parameter values. The lookup-table function reads $\mathcal{T}(a,b) = (c_1(a-\bar{a}) + c_2(b-\bar{b}))^2$ where (c_1,c_2) is the smallest eigenvector in a 2×2 covariance matrix for the matched pixel intensities a and b. The lookup-table function measures the total variance along the direction of the smallest eigenvector, i.e. orthogonal to the direction of the linear mapping between a and b. By minimizing this variance, the correlation coefficient will be maximized.

$$\Delta D_{\text{ORR}} \simeq \sum_{a,b} \Delta h(a,b)(c_1(a-\bar{a}) + c_2(b-\bar{b}))^2 \tag{3.24}$$

3.7 Computing Jacobian Determinants

We will briefly discuss the definition of the Jacobian determinant. This determinant accompanies any coordinate transformation $\boldsymbol{y} = \Psi(\boldsymbol{x})$, with this notation for the three dimensional vectors $\boldsymbol{x} = (x_1, x_2, x_3)$

and $\boldsymbol{y}$ defined as the determinant

$$J_\Psi = \begin{vmatrix} \frac{\partial \Psi_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial \Psi_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial \Psi_3(\boldsymbol{x})}{\partial x_1} \\ \frac{\partial \Psi_1(\boldsymbol{x})}{\partial x_2} & \frac{\partial \Psi_2(\boldsymbol{x})}{\partial x_2} & \frac{\partial \Psi_3(\boldsymbol{x})}{\partial x_2} \\ \frac{\partial \Psi_1(\boldsymbol{x})}{\partial x_3} & \frac{\partial \Psi_2(\boldsymbol{x})}{\partial x_3} & \frac{\partial \Psi_3(\boldsymbol{x})}{\partial x_3} \end{vmatrix} \tag{3.25}$$

The displacement vector field $\Phi_{A\to B}(\boldsymbol{v})$ expresses the transformation $\boldsymbol{v}_B = \boldsymbol{v}_A + \Phi_{A\to B}(\boldsymbol{v}_A)$ from the point $\boldsymbol{v}_A$ in volume A to the point $\boldsymbol{v}_B$ in volume B. The derivatives $\frac{\partial \Psi_{1,2,3}(\boldsymbol{x})}{\partial x_i}$ of the transformation are constant valued between the vector sampling sites when using linear interpolation between vectors. Therefore, we have e.g.

$$\frac{\partial \Psi(\boldsymbol{v})}{\partial v_1} = \frac{\Phi(\boldsymbol{v} + \Delta v_1) - \Phi(\boldsymbol{v})}{\Delta v_1} \tag{3.26}$$

where Δv_1 is the vector sampling distance along the first dimension. In this way, a Jacobian volume is easily computed from the displacement field. The Jacobian 'soundness' of a warp may be evaluated by means of the histogram of all Jacobian's in the displacement field. If the histogram shows an excessive amount of small (less than about one third) or large (larger than about three) values, we have an indication that the warp is under-regularized (:increase α or β). Image slices of the Jacobian can be helpful aids for determining regularization parameters for the warp.

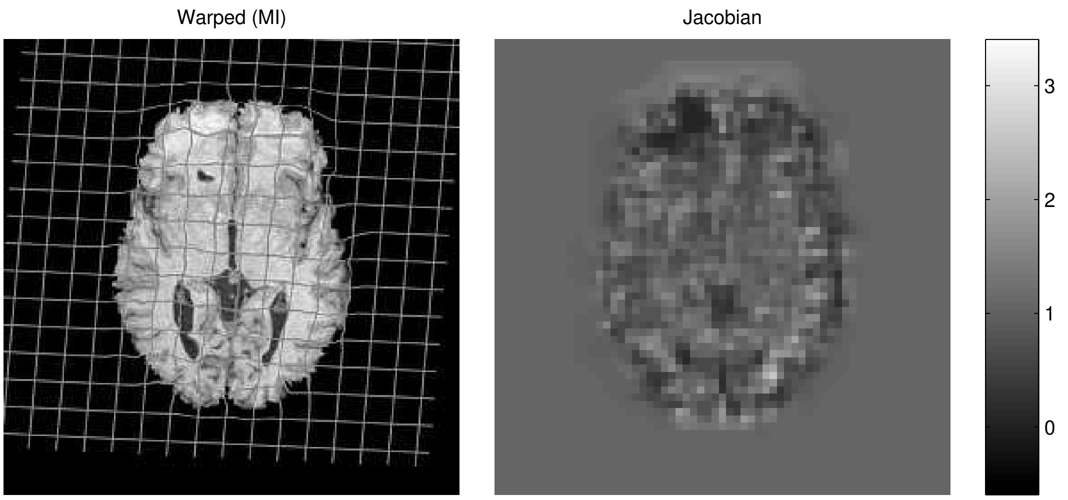


Figure 3.10: Warped slice (using Mutual Information) and corresponding Jacobian values.

3.8 Reconciled Non-linear Registrations

The registration algorithm as described so forth works on a single transformation $A \to B$. Different schemes for regularizing the warp are applied in order to generate 'believable' deformations. In terms of the Jacobian, the regularizer has a somewhat asymmetrical effect, in that large Jacobian values ($J \gg 1$) are penalized much harder than corresponding small values. Consider the one-dimensional example Figure 3.11 below. The Jacobian value $J = 1/4$ is penalized with $9/16$ from the neighborhood penalty term, which is less than the penalty for $J = 2$ which amounts to 1.

The reconciled warp cancels asymmetries in the regularizer by computing the inverse mapping $B \to A$ simultaneously with the mapping $A \to B$, while forcing the two transformations to be close to one another's inverse, i.e. the concatenated transformations $A \to B \to A$ and $B \to A \to B$ should both be *neutral* transformation. This means that a contraction in one transformation is always paired with an expansion in the other transformation. This constraint is implemented into the warping algorithm by running a

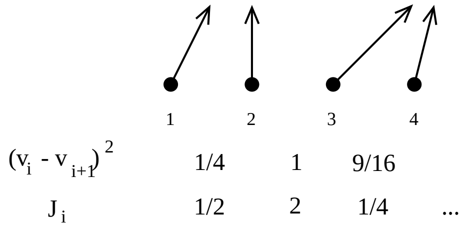


Figure 3.11: Example of asymmetric penalty of quadratic neighbor regularizer. The Jacobian of 1/4 (compression by a factor of 4) is penalized *less* than the Jacobian of value 2 (expansion by a factor of 2).

reconciliation operation in between warp iterations. This operation proceeds as an iterative minimization of the two residuals

$$\varepsilon_{A\to B\to A} = \|\Phi_{BA}(\boldsymbol{v} + \Phi_{AB}(\boldsymbol{v}))\|^2 \tag{3.27}$$

$$\varepsilon_{B\to A\to B} = \|\Phi_{AB}(\boldsymbol{v} + \Phi_{BA}(\boldsymbol{v}))\|^2 \tag{3.28}$$

This is achieved by iteratively updating the $A \to B$ field $\Phi_{AB}^{(q)}$ and $B \to A$ field $\Phi_{BA}^{(q)}$, with q as an iteration index using the equations

$$\Phi_{AB}^{(q+1)} = \frac{\left(\Phi_{AB}^{(q)} - \Phi_{BA}^{(q)}(\boldsymbol{v} + \Phi_{AB}^{(q)}(\boldsymbol{v}))\right) + \Phi_{AB}^{(0)}}{2} \tag{3.29}$$

$$\frac{\left(\Phi_{BA}^{(q)} - \Phi_{AB}^{(q)}(\boldsymbol{v} + \Phi_{BA}^{(q)}(\boldsymbol{v}))\right) + \Phi_{BA}^{(0)}}{2} \tag{3.30}$$

where $\Phi_{AB}^{(0)}$ is the un-reconciled initial displacement field. The iterations are repeated until convergence, which is measured by calculating the mean norm of the change to the displacement vectors. The connection with $\Phi_{AB}^{(0)}$ is needed because the two displacement fields can never exactly fulfill (3.28) unless both fields have zero spatial variation. The solution obtained by the above iterations can be considered a "give and take" solution, both displacement fields will stretch to try to accomplish (3.28), in a sense meeting half way. The modified warp algorithm is outlined in Table 3.4. The reconciliation operations have been built into the warping program, and can be applied in two ways. The program can be either reconcile the forward and inverse displacement fields between every iteration, and the alternative approach is to compute two separate warps and reconcile the output of the individual $A \to B$ and $B \to A$ transformations once afterwards. The latter approach is not recommended though, since the reconciled warp should be a compromise of minimizing the errors $\varepsilon_{B\to A\to B}$ and $\varepsilon_{A\to B\to A}$ while simultaneously minimizing the selected similarity measure.

3.9 Conclusion of Warping Experiment

We have presented a new method for performing fast non-linear registration of MR images. A lookup-table scheme has been presented and demonstrated to be an efficient tool for the implementation and testing of different voxel similarity measures using a model based on the joint probability distribution of matched voxel intensities.

The method has been tested on a functional $[^{15}\mathrm{O}]$water PET experiment and demonstrated to produce a registration superior to a conventional 12-parameter affine registration method. One important improvement was the enhanced registration at the edge of the cortex. This effect is expected to be even more pronounced as the number of subjects in the study increases. Furthermore, a canonical analysis of the functional activation showed that noise was removed from the data: For the non-linear registration methods tested a larger estimated functional "signal-to-noise ratio" was measured: 7.65 for the warp using LS (Scheme C) compared to 5.80 for the 12-parameter normalized data (Scheme B). The experiment described here regrettably does not contain enough subjects to produce reliable error bars on these results—more than 5 subjects are needed to get a proper statistic of the inter-subject variation. Whether or not one can expect similar improvements in functional signal space by applying the registration techniques described here on other datasets presumably depends on factors such as image resolution and noise but also more importantly on the specific activation

```
for resolution = quarter, half, full do

    subsample the template and subject volumes to
    current resolution using Hann weighted sinc
    sub-sampling kernel.

    Interpolate forward and inverse displacement fields
    so that vectors exist for each 4 × 4 × 4 block
    of voxels in the template image/subject image
    respectively.

    repeat

        Compute cost function look-up tables
        T_AB(a,b) and T_BA(a,b) from current histogram
        h^n_AB(a,b) and h^n_BA(a,b).

        Update elements Φ_ijk for which i+j+k is
        odd maximizing the estimated decrease in the
        global cost function
```

$$\Phi^n_{ijk} = \underset{\phi \in S_{ijk}}{\text{minarg}} \; \Delta\hat{C}_\phi$$

```
        Update elements Φ_ijk for which i+j+k is
        even.
        Repeat updates for inverse mapping, now
        using h^n_BA(a,b) and T_BA(a,b).

        Apply low-pass filter^5 to Φ^n_AB and Φ^n_BA.

        Reconcile forward and inverse displacement
        fields

        Compute new histogram of matched image
        intensities h_AB(a,b), h_BA(a,b), and compute
        the global similarity measures from this.

    until (selected similarity measure has converged in
    both transformations)

end for
```

Table 3.4: The outline of the reconciled warping algorithm computing a forward and inverse mapping simultaneously.

paradigm. Most of the activation hot-spots are handled quite well by the well-known 12 parameter affine registration methods and we have also demonstrated that there was no advantage in using matched MRI scans with the simpler affine registration procedure. If the paradigm in question generates activation in cortical areas that are already handled quite well by the simpler methods there may be little advantage in introducing the overhead of non-linear registration techniques.

Conclusion

We have explored the mathematical foundations of image noise reduction focusing on the relation between the statistical mechanics approach and the traditional linear theory. Useful image processing tools based on random fields were demonstrated on functional and anatomical scans of the human brain; techniques which were shown to have a high degree of flexibility in the design.

The linear theory of AR modeling was shown to have a direct relation to parameter estimation in Markov random field models. This relation causes the Gaussian Markov Random field models to inherit the stability problems of all-pole modeling: simply going for the Maximum Likelihood estimate will typically be a sub-optimal approach for non-causal image models. Care must be taken when defining the parameter space for such models. A Bayesian analysis of maximum likelihood based parameter estimation suggested that the 'native' form of parameter space, e.g. a log-transform, may in some cases lead to more well-behaved maximum likelihood parameter estimators.

Maximum likelihood estimation in the broad class of Gibbs distributions is viable through the generalized Boltzmann Learning rule. Modifications of the learning rule for proper treatment of scale parameters were presented. The analytical treatment of the learning rule continued with the derivation of rule for learning parameterized covariance structures. The attempt to make the 'ends meet' and perform AR modeling from an iterative power spectrum learning rule derived from the Boltzmann learning rule revealed stability problems originating from the poles in the fitted spectrum. Methods for avoiding these stability problem by the use of scale parameters were considered.

A new algorithm for performing fast non-linear registration of MR images was presented. The algorithm comprised an optimized look-up table scheme which made it possible to compare different voxel similarity measures based on the joint distribution of matched voxel intensities. A practical application of the algorithm to a saccadic eye movement study revealed superior performance of the algorithm compared to a conventional 12-parameter affine registration method. In addition, a canonical analysis of the functional activation patterns showed that noise was removed from the data as a direct result of the improved registration: the signal to noise ratio was increased from 5.80 using 12 parameter affine registration based on PET alone, to 7.65 for the warp using a Least Squares similarity measure. A novel improvement for the registration algorithm was presented using a reconciliation procedure for simultaneous computation of a forward and an inverse displacement field.

Appendix A

Publications

The following pages reproduce publications of the last three year's work.

Revealing Structural Effects In Functional Imaging with Anatomical Warps

U. Kjems[1], C.T. Chen[2], S.C. Strother[3,4], L.K. Hansen[1], J.R. Anderson[3], I. Law[5], O.B. Paulson[5], I. Kanno[6], D.A. Rottenberg[3,4]

[1]Department of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark
[2]Department of Radiology, University of Chicago, Illinois, Chicago, USA
[3]Veterans Administration Medical Center (VAMC) and [4]University of Minnesota, Minneapolis, USA; [5]Deparment of Neurology, Rigshospital (RH), Copenhagen, Denmark
[6]Research Institute of Brain and Blood Vessels, Akita (ARI), Japan

Introduction. In functional activation studies, understanding and compensating for intersubject differences may be important in identifying reproducible activation patterns. For a simple motor task the distribution of [^{15}O]water PET activation images from different sites (ARI, RH, VAMC) contains significant intersubject variation including site-specific oriental vs. occidental clustering. Our long term goal is to determine whether this clustering is due purely to methodological and/or anatomical intersubject variation or represents true racial functional differences. As a step towards this goal we have developed a new non-linear anatomical warp algorithm which will allow us to investigate the impact of anatomical differences on intersubject variation.

Methods. [^{15}O]water PET scans were obtained from 10 subjects (10 scans/subject; 2 Japanese at ARI; 4 Danish at RH; 4 American at MVAMC) performing a left-handed sequential finger-to-thumb opposition task. Scans were aligned (AIR, 1) and analyzed to obtain the eigenvectors from a Scaled Subprofile Model with principal component analysis (SSM/PCA, 2). In addition, MR scans were collected from an oriental and an occidental normal volunteer in order to test the new warp algorithm, which is a computationally optimized 3D method related to the 2D technique described in (3); it utilizes an iterative search scheme to generate a displacement field that can be applied to warp a test image to match a target image. The information obtained from warping the anatomical MR images can be applied to matched functional images in order to reduce functional intersubject variation due to structural differences.

Results. Figure 1 (left panel) plots the distribution of scan weights for the first two eigenvectors from the SSM/PCA analysis of the 10 subjects. Clear site-specific clustering is seen demonstrating oriental vs. occidental differences (ARI vs. RH-VAMC), which are captured by the distribution of scan weights for eigenvector 2. These results demonstrate that intersubject variation between sites is much larger than intersubject variation within sites and may be partly dependent on anatomical and/or functional racial differences. Fig. 1 (right panel) illustrates a preliminary example of anatomical warping of an oriental (test image) to an occidental (target image) brain. The non-linear warped images (lower panels) match the target better than those produced by the linear warp (upper panels; AIR, 1).

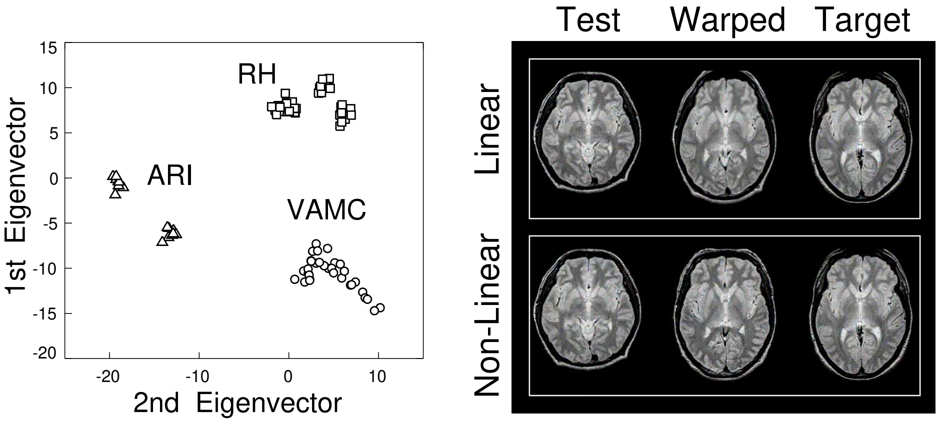


Figure 1. Site-specific neuroimage clustering and oriental-occidental warping

Conclusion. We hypothesize that a nonlinear anatomical normalization, such as that provided by our new warping algorithm, may be used to reduce the site-specific structure found in multi-site functional datasets. Patterns of intersubject variance from SSM/PCA may be used as a metric to assess the performance of nonlinear warping algorithms and, in addition, to assess the use of nonlinear warps to separate inter-racial structural differences from methodological and functional effects.

References

1. Woods RP, Mazziotta JC, Cherry SR. In: Proc. Brain PET'93 Akita:Quantification of Brain Function (Uemura K, et al., eds.), Amsterdam, Excerpta Medica, 1993, pp. 391-400
2. Strother SC, Anderson JR, Schsper KA, et al. J. of Cereb. Blood Flow Metab., 1995, 15:738-753
3. Kosugi Y, Sase M, Kuwatani H, et al. J. Comp. Assist. Tomogr., 1994, 17:634-640

1

A New 3D Non-linear Brain MRI Registration Algorithm Improving Functional $\left[^{15}\mathrm{O}\right]$water PET Registration

Ulrik Kjems, Stephen C. Strother, Jon Anderson,
Ian Law, Lars Kai Hansen

Abstract— **This paper addresses the problem of anatomical registration across individuals for functional $\left[^{15}\mathrm{O}\right]$water PET activation studies. A new algorithm for 3D non-linear structural registration (warping) of MR scans is presented. The method performs a hierarchically scaled search for a displacement field maximizing one of several voxel similarity measures derived from the two dimensional histogram of matched image intensities subject to a regularizer that ensures smoothness in the displacement field. The effect of the non-linear structural registration is studied when it is computed on anatomical MR scans and applied to co-registered $\left[^{15}\mathrm{O}\right]$water PET scans from the same subjects; in this experiment a study of visually guided saccadic eye movements.**

The performance of the non-linear warp is evaluated using multivariate functional signal and noise measures. These measures prove to be useful for comparing different inter-subject registration approaches, e.g. affine versus non-linear.

A comparison of 12-parameter affine registration versus nonlinear registration demonstrates that the proposed non-linear method increases the number of voxels retained in the cross-subject mask. Furthermore we demonstrate that improved structural registration may result in an improved multivariate functional signal-to-noise ratio. It is further demonstrated that registration of the PET scans using the 12-parameter affine transformations that align the co-registered MR images does not improve registration compared to using 12-parameter registration on the PET images directly.

Keywords— **Non-linear warping, stereo-tactic registration, inter-subject registration, voxel similarity measures.**

I. Introduction

When investigating functional activation patterns in multiple subjects, a basic problem is that of comparing and averaging the functional activation images across subjects. Apart from rigid body translation and rotation, the inter-subject registration problem is essentially how to find a three dimensional deformation of a subject's anatomy to match a template anatomy. The deformation ideally represents a *correspondence map*, between the observed image and the template image, to the extent that such a map exists.

Multiple scans from one or more sensor modalities of the same subject may be automatically registered quite accurately using rigid body rotations and translations, involving 6 parameters in the transformation procedure. In [1] it is shown how intra-subject registration MRI-MRI and PET-MRI can be achieved with good accuracy across the entire volume (~1–2 mm) using the Automated Image Registration (AIR) method as described in [2], [3], [4], [5].

In contrast to this, inter-subject registration is more complex in terms of the number of parameters necessary: Ultimately, the correspondence map consists of three parameters for each voxel, representing the x, y and z location of the corresponding voxel in a template image typically defined in the standard Talairach stereotactic space [6]. Automated inter-subject registration procedures typically overcome this parameter estimation problem by parameterizing the deformation field in a lower dimension. An example of this is the 12-parameter transformation supported by the AIR package [2], [3], [4], [5] which recently has been extended to include 5th order polynomial transformations with up to 168 parameters.

This paper compares the resulting functional activation patterns achieved when applying non-linear registration techniques across subjects to those achieved using conventional 12-parameter affine[1] transformations. The inter-subject functional variation (e.g. the variance across subjects in $\left[^{15}\mathrm{O}\right]$water PET studies) is composed of a mixture of measurement noise, anatomical variation and functional variation, e.g. variation in how a given subject's activation pattern is linked to the anatomy. Since the inter-subject structural registration is limited to a certain spatial scale, we need to choose the degree to which the structural registration should be performed. The image resolution together with the functional variation between subjects set the lower boundary on the scale at which the anatomical registration will be useful. It may be that the effort of performing a detailed non-linear structural registration is not rewarded by an increase in the extracted functional activation signal, because the non-linear registration operates at a smaller scale than the functional differences that are known to exist within a set of normal subjects. We wish to know whether the proposed non-linear registration algorithm leads to an "improvement" in the measured brain

U. Kjems and L. K. Hansen are with the Department of Mathematical Modelling, Technical University of Denmark. E-mail: uk@imm.dtu.dk

S. C. Strother and J. Anderson are with the PET Imaging Service, VA medical center, Minneapolis. S. C. Strother is also with the Radiology and Health Informatics Depts., University of Minnesota, Minneapolis

I. Law is with the Department of Neurology, National University Hospital of Denmark, Copenhagen

[1] An affine transformation is one that preserves straight parallel lines. It is conveniently expressed as a matrix multiplication of a 4 by 4 transformation matrix with a homogeneous coordinate vector [7].

activity map, as opposed to using a simpler conventional method, e.g. using 12-parameter affine transformations.

In the non-linear registration procedure presented different voxel similarity measures may be applied and it is not evident beforehand which of these measures is the optimal. Furthermore, the algorithm uses a set of regularizing parameters, which control the smoothness of the displacement field that is obtained, and we need to consider how to find an optimal setting for such parameters.

In order to address the above problems, we have evaluated the anatomical registration by examining how well the corresponding functional data have been registered and how the functional activation signal is improved. It is assumed that the optimal registration is the one that makes the functional activation signal stand out most clearly from the baseline condition for a group of subjects. Using this evaluation approach we are able to compare different registration techniques, voxel similarity measures and regularization parameter settings.

The rest of this paper is arranged as follows. The non-linear registration algorithm is described in section II. In Section III we describe the functional data set used for the alignment experiments, and in Section IV the analysis methods used for examining the functional signal space are described and the results of the analysis are discussed.

II. The Non-linear Registration Algorithm

The method described here is based on a 3D extension of the approach described in [8] although several extensions have been added. The method is also similar in construction to the ANIMAL algorithm developed by Collins *et al.* [9], [10] and the methods described in [11], [12]. A considerable amount of work exists based on non-linear deformations of human brain anatomy—most of the work has been done with the approach of investigating the anatomical variation between individuals [13], [14]. But few authors have investigated to what extent functional inter-subject variation in brain volumes can be corrected for using the anatomical information [15].

The major difference between this method and previously published methods lies in the regularization applied and in the evaluation of the voxel similarity measures. Instead of optimizing the similarity measure locally, this method performs the optimization based on the change in the global similarity measure subject to regularizing constraints. Furthermore, the method is optimized to yield fast convergence, so that a full non-linear registration of a high-resolution $256 \times 256 \times 160$ volume with a $64 \times 64 \times 40$ vector field can be computed in 25 minutes on a 90MHz R4400 processor. The techniques for evaluation of global change of similarity measures presented below may also be adapted to warp-algorithms that apply different strategies for searching for the optimal displacement field, e.g. the wavelet based method of [16].

A volume of displacement vectors Φ overlaying the template volume is computed. These vectors are sampled on a (typically 4 times) coarser grid and represent local displacements that make the subject image volume similar to the template image. The vector field is computed in a hierarchical manner, by first iterating the displacement field using a coarsely sampled vector grid on sub-sampled instances of the subject and template volumes, and increasing resolution whenever the similarity measure has converged until the original resolutiuon of the MR volumes is reached. Once the set of displacement vectors is computed, the actual displacement of each voxel in the high-resolution volume may be found by tri-linear interpolation between the 8 local surrounding vectors on the (coarser sampled) grid of displacement vectors.

A. The Algorithm

The details of the algorithm are explained using a pseudo-C description in Table I. We have had stable results when the vector field is four times coarser than the template image, i.e. each vector is associated with the $4 \times 4 \times 4$ cube of nearest voxels. Three resolution steps have been sufficient for all scans tested when the scans have been prealigned using a 12-parameter affine transformation prior to the non-linear warp. We have used our own implementation of the AIR method for this step.

Let $\mathcal{A}(\boldsymbol{v})$ and $\mathcal{B}(\boldsymbol{v})$ be the template and subject image volumes, respectively, indexed by the vector $\boldsymbol{v} = (p, q, r) \in \mathcal{G}$, consisting of discretely sampled and (8 bit integer) quantized voxel intensities a and b. The deformation field at iteration n, denoted Φ^n, consists of vectors Φ^n_{ijk} with the coarser grid indexed by (i, j, k). The template voxel $\mathcal{A}(\boldsymbol{v})$ is subject to a shift in coordinates so that $\mathcal{A}(\boldsymbol{v})$ is mapped to $\mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}})$, where $\Phi^n_{\boldsymbol{v}}$ is obtained by trilinear interpolation of the 8 nearest displacement vectors on the coarser sampled grid of vectors Φ^n_{ijk} as shown in Figure 1.

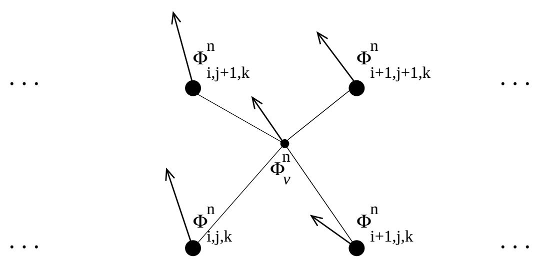


Fig. 1. Interpolation of displacement field vectors. The notation $\Phi^n_{\boldsymbol{v}}$ implies a tri-linear interpolation of the 8 surrounding grid vectors Φ^n_{ijk} to find the displacement for the voxel at location $\boldsymbol{v}$.

Different assumptions on the exact nature of this mapping may be made depending on the sources of image volumes $\mathcal{A}$ and $\mathcal{B}$. If the scans are recorded using identical acquisition sequences on the same scanner, it may be fair to assume a linear mapping of the voxel intensities e.g. the correlation voxel similarity measure. Such an approach may not be viable if the tissue contrasts differ. In such cases a more general mapping should be assumed—e.g. the mutual information (MI) similarity measure. MI has gathered considerable interest for inter-modality image registration, see e.g. [17], [18], [19], [20], [21]. Variations in tissue contrast *between* the volumes should not be confused with variation *within* a single volume typically caused by field inhomogeneities—such effects are presumed to have been corrected for prior to the registration process.

TABLE I
THE OUTLINE OF THE WARPING ALGORITHM.

```
for resolution = quarter, half, full do

        subsample the template and subject volumes to
        current resolution using Hann weighted sinc
        sub-sampling kernel.

        Interpolate displacement field so that vectors
        exist for each 4 × 4 × 4 block of voxels in the
        template image.

        repeat

                Compute cost function look-up table T(a,b)
                from current histogram h^n(a,b).

                Update elements Φ_ijk for which i+j+k is
                odd maximizing the estimated decrease in the
                global cost function

                        Φ^n_ijk = minarg_{φ ∈ S_ijk} ΔĈ_φ

                Update elements Φ_ijk for which i+j+k is
                even.

                Apply low-pass filter to Φ^n.

                Compute new histogram of matched image
                intensities h(a,b), and compute the global
                similarity measures from this.

        until (selected similarity measure has converged)

end for
```

A.1 Iteration Procedure

At each iteration step, vector estimates are updated by searching for the largest possible estimated decrease in a global cost function, which is assumed to take the form

$$C(\Phi^n) = D\big\{\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}})\big\}_{\boldsymbol{v}\in\mathcal{G}} + R\big(\Phi^n\big), \quad (1)$$

with $D(a,b)$ being a function measuring the similarity of a set of matched voxel intensities a and b, and $R(\Phi^n)$ is a regularizing constraint representing our prior knowledge of the nature of the displacement field, i.e. that it is smooth. We have applied a simple Gaussian Markov random field [22] prior consisting of the sum of squared differences of the 6 nearest neighboring vectors indexed by $i'j'k' \in \mathcal{N}_{ijk}$

$$R(\Phi^n) = \frac{\alpha}{2}\sum_{ijk}\sum_{i'j'k'\in\mathcal{N}_{ijk}}\left\|\Phi^n_{ijk} - \Phi^n_{i'j'k'}\right\|^2, \quad (2)$$

introducing the global regularization parameter α. In the following we shall assume that D can be expressed as a function of the probability distribution of matched voxel intensities $p(a,b)$. The estimated change of $C(\Phi^n)$ from moving a single displacement vector Φ^n_{ijk} to ϕ is computed by a linear approximation. The notation $C^n_{\phi,ijk}$ denotes the value of the cost function (1) when replacing the single vector Φ^n_{ijk} by ϕ.

$$\Delta C^n_{\phi,ijk} \simeq \Delta R(\Phi^n) + \sum_{a,b}\Delta p(a,b)\frac{d}{dp(a,b)}D(\Phi^n). \quad (3)$$

The distribution $p(a,b)$ may be estimated directly from the two-dimensional discrete histogram $h^n(a,b)$ given the current estimate of the displacement field:

$$h^n(a,b) = \sum_{\boldsymbol{v}\in\mathcal{G}}\delta\big\{\mathcal{A}(\boldsymbol{v}) - a, \mathcal{B}(\boldsymbol{v} + \Phi^n_{\boldsymbol{v}}) - b\big\} \quad (4)$$

$$p(a,b) \simeq h^n(a,b)/N, \qquad N = \sum_{a,b} h^n(a,b) \quad (5)$$

A limited neighborhood centered around the previous iteration vector is searched exhaustively evaluating equation (3) at integer displacements of the corresponding block of template voxels. The neighborhood is limited to ± 2 voxels shift along the 3 axes giving a total of 125 search locations for each individual vector update.

$$\Phi^{n+1}_{ijk} = \underset{\phi\in S^n_{ijk}}{\text{minarg}}\ \Delta\hat{C}_\phi \quad (6)$$

$$S^n_{ijk} = \Big\{\phi \in \mathcal{Z}^3 \,\Big|\, |\phi - \Phi^n_{ijk}|_\infty \le 2\Big\} \quad (7)$$

A simple estimate of the change of the probability distribution $p(a,b)$ caused by replacing Φ^n_{ijk} with ϕ can be obtained by assuming that the box of voxels associated with the vector Φ^n_{ijk} has a constant change of displacement. Let Q_{ijk} represent the set of coordinates indexing the template box of voxels associated with vector Φ_{ijk} so we can write

$$\Delta p_\phi(a,b) \simeq \frac{1}{N}\sum_{\boldsymbol{v}\in Q_{ijk}}\delta\big\{a - \mathcal{A}(\boldsymbol{v}), b - \mathcal{B}(\boldsymbol{v}+\phi)\big\} - \frac{1}{N}\sum_{\boldsymbol{v}\in Q_{ijk}}\delta\big\{a - \mathcal{A}(\boldsymbol{v}), b - \mathcal{B}(\boldsymbol{v}+\Phi^n_{ijk})\big\}$$

Using this result we get

$$\begin{aligned}\Delta D_\phi &\simeq \sum_{a,b}\Delta p_\phi(a,b)\frac{dD(\Phi^n)}{dp(a,b)}\\ &\simeq \sum_{\boldsymbol{v}\in Q_{ijk}}\frac{1}{N}\left\{\frac{dD(\Phi^n)}{dp(\mathcal{A}(\boldsymbol{v}),\mathcal{B}(\boldsymbol{v}+\phi))} - \frac{dD(\Phi^n)}{dp(\mathcal{A}(\boldsymbol{v}),\mathcal{B}(\boldsymbol{v}+\Phi^n_{ijk}))}\right\}\\ &= c + \frac{1}{N}\sum_{\boldsymbol{v}\in Q_{ijk}}\mathcal{T}\big\{\mathcal{A}(\boldsymbol{v}),\mathcal{B}(\boldsymbol{v}+\phi)\big\} \quad (8)\end{aligned}$$

$$\mathcal{T}\{a,b\} = \frac{d}{dp(a,b)}D^n_\Phi \quad (9)$$

where c is independent of Φ^{n+1}_{ijk}. Equation (9) first applies a first order approximation[2] to the change in ΔD_ϕ

[2]It should be noted that the first order approximation poses a problem if D is not differentiable at $p(a,b) = 0$. We shall address this problem in Section II-A.2.

wrt. $p(a,b)$, and secondly the aforementioned approximation to the change $\Delta p(a,b)$. Thus, the final change in D is estimated by summing values from a look-up table $\mathcal{T}(a,b)$ which can be computed from the current estimate of the $p(a,b)$ distribution, and has a size proportional to the square of the number of discrete levels that a and b may take, i.e. 256×256 if 8 bit internal representation is used. This look-up strategy makes the optimization of the global similarity measure computationally feasible since the bulk of computational work is reduced to indexing and summation operations. An update of an individual vector consists of, among all $\phi \in S_{ijk}$, choosing the one that minimizes the cost function, i.e. minimizing

$$\begin{aligned} \Delta\hat{C}_\phi &= \frac{1}{N}\textstyle\sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{T}\{\mathcal{A}(\boldsymbol{v}), \mathcal{B}(\boldsymbol{v}+\phi)\} \\ &+ \alpha\textstyle\sum_{i'k'j'\in\mathcal{N}_{ijk}} \left\|\phi - \Phi^n_{i'j'k'}\right\|^2 . \end{aligned} \tag{10}$$

The vectors are updated in a 3D checker-board fashion, i.e vectors Φ_{ijk} for which $i+j+k$ is even are updated in the first round and the remaining vectors in the second round. This is done to increase the convergence speed of the cost function (1). Equation (10) shows that a vector update has a Markov property that each update depends only on the 6 nearest vector neighbors—this means that by updating first one half of the 3D checker-board and then the other half, the second half of updates will be experiencing the neighbors Φ^{n+1} field rather than the Φ^n field. This will in effect double the convergence rate.

Finally, the displacement fields are smoothed in between iterations by applying a 3D Gaussian spatial low-pass filter. The size of the Gaussian kernel FWHM, which we will denote β, is a smoothness parameter that needs to be carefully set together with the α of Equation (2).

A.2 Similarity Measures

Three different similarity measuring functions are described here, least squares (LS), correlation coefficient and the information based mutual information (MI). The look-up table $\mathcal{T}(a,b)$ is computed at each iteration step as listed in Table II. In case of the correlation and MI measures, $p(a,b)$ must be estimated between iterations, which involves the overhead of computing $\mathcal{B}(\boldsymbol{v}+\Phi^n_{\boldsymbol{v}})$. For the LS measure, $\mathcal{T}(a,b)$ may be computed on line, i.e. with no table actually computed which makes the LS method faster.

The LS using the equation in Table II is quite sensitive to global intensity variations in the MR scans. The LS algorithm referred to in the second part of the paper is actually a modified LS: The estimated change in the distance measure by replacing Φ^n_{ijk} with ϕ can be obtained in line with the approach used in Equation (9) using

$$\Delta D_\phi \simeq \sum_{\boldsymbol{v}\in Q_{ijk}} \frac{1}{N}\left\{\mathcal{A}(\boldsymbol{v}) - \bar{\mathcal{A}} - \mathcal{B}(\boldsymbol{v}+\phi) + \bar{\mathcal{B}}\right\}^2 \tag{11}$$

where $\bar{\mathcal{A}} = \sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{A}(\boldsymbol{v})/N_Q$ and $\bar{\mathcal{B}} = \sum_{\boldsymbol{v}\in Q_{ijk}} \mathcal{B}(\boldsymbol{v} + \Phi^n_{ijk})/N_Q$. This makes the algorithm fit the edges rather than the local intensity level of the data.

The estimation of $p(a,b)$ using (5) poses a problem in the case of MI. It is evident from Table II that $\mathcal{T}(a,b)$ breaks down in the neighborhood of $p(a,b)=0$. Increasing the bin size of the histogram $h^n(a,b)$ so that no bin gets a zero count would solve the problem but throws away intensity resolution. Increasing the bin-size can be thought of as smoothing the histogram with a box-filter, so instead of using the same filter size all over the histogram, an adaptive smoothing is performed. Areas of $h^n(a,b)$ with low counts are smoothed with a large scale kernel (similar to a large bin), areas with higher counts with a smaller kernel. The kernel used is a Gaussian kernel with a FWHM adjusted so that the signal to noise ratio of the smoothed histogram counts is approximately constant, assuming local uniformity of $p(a,b)$. The signal to noise ratio is computed from the variance in the binomial distribution: Let $h(a,b)$ be the current histogram of matched intensities and g_f a Gaussian blurring kernel of FWHM equal to f. Assuming local uniformity of $h(a,b)$ we may estimate a local signal to noise ratio (SNR) using

$$\begin{aligned} \text{SNR} &\simeq \frac{h(a,b)}{\sqrt{\operatorname{Var}\{h * g_f(a,b)\}}} \\ &\simeq \frac{h(a,b)}{\sqrt{\sum_{a',b'} g_f^2(a',b')h(a-a',b-b')}} \\ &\simeq \sqrt{\frac{\pi h(a,b) f^2}{2\log 2}} \end{aligned} \tag{12}$$

so that

$$f \simeq \text{SNR}\cdot\sqrt{\frac{2\log 2}{\pi h(a,b)}} \tag{13}$$

In the actual implementation the histogram is smoothed with 5 blurring kernels (FWHM = 2, 3, 5, 7 and 10 histogram indices). For each (a,b), Equation (13) selects the correct local blurring size among these 5 histograms. In order to stabilize the procedure, a blurred histogram (FWHM=2) is used for $h(a,b)$ in (13). A SNR value of 10.0 produced good results. After applying this scheme, no bins will contain the zero count value (because of the global range of the Gaussian kernel), and the parts of the histogram that have a high number of counts preserve their information.

A.3 Discussion

Both the sum of squares and the correlation similarity measures rely on the fundamental assumption that the residual difference between the warped subject image and the template is white Gaussian noise. This can be realized by thinking of the cost function (1) as a negative log posterior distribution. The regularizer R is a prior distribution on the displacement vectors while the similarity measuring term D is a model of the residual noise, which in the quadratic cases is a product of independent Gaussian distributions. We know, however, that this assumption is incorrect in the sense that the subject anatomy *is not* a

TABLE II

LOOKUP-TABLE FUNCTIONS $\mathcal{T}(a,b)$ DERIVED FROM 3 DIFFERENT SIMILARITY MEASURES D. NOTE THAT CONSTANT TERMS AND SCALING FACTORS ARE IGNORED IN THE $\mathcal{T}(a,b)$'S. ALSO NOTE THAT THE D LISTED FOR MI IS ACTUALLY THE NEGATIVE MUTUAL INFORMATION—THE WARP ALGORITHM MINIMIZES D, THEREBY *maximizing* THE MUTUAL INFORMATION FOR $p(a,b)$. (c_1, c_2) IS THE EIGENVECTOR WITH SMALLEST EIGENVALUE OF THE 2×2 COVARIANCE MATRIX BETWEEN INTENSITIES a AND b.

Method	Distance measure	$\mathcal{T}(a,b)$
Least Squares	$\sum_{a,b} p(a,b)(a-b)^2$	$(a-b)^2$
Correlation Coefficient	$\rho_{ab} = \frac{\sqrt{\sum_a p(a)(a-\bar{a})^2 \sum_b p(b)(b-\bar{b})^2}}{\sum_{a,b} p(a,b)(a-\bar{a})(b-\bar{b})}$	$(c_1(a-\bar{a}) + c_2(b-\bar{b}))^2$
Mutual Information	$-\sum_{a,b} p(a,b) \log \frac{p(a,b)}{p(a)p(b)}$	$-\log \frac{p(a,b)}{p(a)p(b)}$

deformed instance of the template anatomy—real anatomical differences certainly exist among normal brains. The major disadvantage of the squared similarity measures is that the anatomical differences that cannot be corrected for by the warp are penalized with a factor depending on how the image modality represents the mismatched tissue types as image intensities. There is no reason to believe that a mis-registration of e.g. gray matter with CSF should be penalized by a factor depending only on the difference in the intensity representations of those two tissue types. The MI on the other hand, penalizes (mis)matches with a factor depending on the resulting specificity (i.e. entropy of the conditional distribution $P(b|a)$ of subject voxel intensities b given corresponding template voxel intensity a) of the mapping between image intensities. This is certainly a more general approach, and it may be a better criteria.

Another relevant issue is the number of iterations needed for the warp to converge at each resolution step. Our experience has been that MI requires twice the number of iterations to converge (typically 8) compared to the sum of squares method (typically 4) so that the total computation time is approximately doubled. This is because the probability distribution $p(a,b)$ is a more complex function of the displacement field when using MI than LS or correlation, so that more iterations are needed in each resolution step.

A.4 Smoothness Constraints

Several factors are working together to ensure the smoothness of the displacement field. The reason for requiring smoothness is simply a question of controlling noise in the resulting displacement map. The factors are:

- *Use of a regularizer, or prior term in the global cost function (1).* The term $R(\Phi)$ is essentially a Gaussian Markov field prior. The frequency power spectrum for this prior may be calculated analytically, see e.g. [23]. Previous work has been published applying physical/mechanical prior distributions to the deformation problem [12], [24], [25], [26], [27], [28]. However, considering that the deformation itself is non-physical (i.e. the subject anatomy *is not* a deformed instance of the template anatomy), we argue that neither does the prior distribution have to be.
- *Controlling the sampling of the displacement field.* By limiting the number of vectors in the representation of Φ we efficiently limit the upper frequency components of the deformations that can be represented. In [29], the displacement field is expressed as an expansion of smooth basis functions thereby restricting the resolution of the resulting displacement field. In [16] a hierarchical search for the displacement field is performed by representing the displacement field in wavelet components, and gradually increasing the number of components used.
- *Smoothing of the image volumes.* This technique is used implicitly in the sub-sampling of the subject and template volumes. It is also used in [29] and in the ANIMAL algorithm described in [9], [10], [13], [30].
- *Direct smoothing of the vector field components.* This approach is a direct way of shaping the frequency characteristics of the displacement field. The parameter β controls the size of the Gaussian low-pass filter used. Low-pass filtering of the displacement field vectors is also used in [8].

The problem of actually choosing the parameters used is a topic of current investigation. At each resolution step a new set of parameters is applied. Preliminary work based on the effect of the warp on the analysis of a functional data set has indicated that α and β cannot substitute for one another, although there exists a correlation in the optimal (with respect to the resulting functional alignment as described in Section IV) setting of the two parameters. It should further be noted that the setting of α depends strongly on the voxel similarity measure used, since the effect of the regularizing term changes with the dynamics of the match term in (1).

III. DATA ACQUISITION

The functional activation study was a saccadic eye movement experiment [31] done at Rigshospitalet, Copenhagen, Denmark, consisting of 5 male right-handed subjects, each scanned 8 times (2 baseline, 6 graduated target frequency) during two scanning sessions, over two days with randomized ordering of scans. An array of LED's was located in an arc at angles -40° to 40°. The subjects performed visually guided voluntary anti-saccades. This paradigm involved subjects fixating on a central LED until a random target LED was lit. The task was to perform a saccade to the LED contralateral to the lit target. The target presentation frequency was varied from scan to scan from 0.05, 0.1, 0.3, 0.5, 0.7 to 0.9 Hz. Two baseline scans were acquired with the subject fixating at the central LED. The scanner is a whole body PET-scanner (Advance, General Electric, [32]) with an image resolution of approximately 5 mm in all directions. The scans were reconstructed with 35 slices, voxel size $2.0 \times 2.0 \times 4.25$ mm covering a field of view of $25.6 \times 25.6 \times 15.2$ cm. In addition, MR scans were obtained for the 5 PET subjects plus an additional subject to act as an alignment template. The scans were T1-weighted acquisitions using a Siemens Magnetom Vision scanner located at Hvidovre Hospital, Copenhagen, with $0.98 \times 0.98 \times 1.17$ mm voxel size for two of the five subjects and $0.98 \times 0.98 \times 1.00$ mm voxel size for the remaining 4 subjects, all acquired using a 3D MPRAGE sequence. The PET experiment consisted originally of 8 subjects out of which 2 did not wish to have the MR scan, and 1 subject was removed because his MRI showed an abnormality in the cerebellum, leaving 5 subjects for this experiment.

A. Preprocessing

A common preprocessing step was performed on all scans. Each subject's PET scans were intra-subject aligned using a 6 parameter rigid body rotation and translation (AIR, [3]) to match each scan to the subject's first scan. A mean scan was computed for each subject and a slice based threshold of 45% of the slice maximum was applied. The injected dose/weight ratios were not available, so global scaling of each scan was achieved using a voxel based Scaled Subprofile Model (SSM) analysis ([33], [34], see Section IV). No spatial smoothing was applied to the scans following the reconstruction, yielding an effective resolution of approximately 5 mm.

The MR scans were corrected for intensity inhomogeneity, which was predominant in the axial (Z) direction. This was accomplished by computing the mean of the top 5% intensity voxels for each slice containing brain tissue, and normalizing the slice intensities by fitting a 6'th order polynomial through these values as a function of the slice number, which gave a visually improved result. The MR scans were then carefully stripped from skull and scalp using a manual interactive stripping/region drawing tool, retaining mainly CSF, white matter and grey matter within a contiguous mask.

B. Three Alignment Schemes

Three registration schemes were set up for comparison as illustrated in Figure 2. Scheme A was our standard alignment procedure using a 12-parameter affine registration to align the mean PET images for each subject with a Talairach PET template volume obtained by simulating a PET scan from a high-resolution segmented MRI scan [1]. In Scheme B the 12-parameter inter-subject registration was computed using the subject's stripped MR volumes and a Talairach transformed MR template brain, and the PET images were aligned via this transformation, so that each PET scan was registered by combining the individual 6 parameter PET-PET and PET-MR transformations, and 12-parameter affine transformations in a single resampling operation. In Scheme C–E a 3D non-linear registration field was computed on each of the subject's 12-parameter registered, stripped MR scans to match the stripped template scan. Each PET scan was resampled combining the non-linear warp, the individual 6 parameter PET-to-PET, PET-to-MR, and 12-parameter MR to template transformations in a single resampling operation.

In all three cases, both the raw PET images and the binary masks (1 is tissue, 0 is background) generated from the mean PET images were resampled using tri-linear interpolation in the same resolution as the original data ($2 \times 2 \times 4.25$ mm). New binary subject masks were formed by thresholding the resampled masks at 0.5. All 5 resampled masks were then logically AND'ed to find the resulting global mask used for the functional analysis.

Apart from the absence of a step with spatial smoothing of the PET images, Scheme A was identical to our standard procedure used for group analysis of PET activation studies also described in [1], [35]. This is a relevant comparison since in many cases, MRI scans are not acquired routinely with the PET scans.

One may argue that method B and C are more sensitive to errors in the 6 parameter registration than method A since they concatenate two rigid body transformations while method A only involves one. This suggests the alternative procedure of aligning each of the subject's PET scans to the MR scan individually. That approach has not been used because it is our experience that the noise in the raw PET scans gives problems with the cross modality registration, so that the total registration error becomes larger. A possible solution to this dilemma for future investigation is offered in [4] by reconciling all possible pairwise rigid body transformations using an iterative optimization scheme.

For comparison, three similarity measures for the non-linear warp were selected: The modified least square (LS), correlation coefficient (CC) and mutual information (MI), producing three separate non-linearly aligned data sets (which we shall refer to as C, D and E, respectively). At this point we face the problem of setting the regularization parameters α Equation (10) and β of Section II-A.1. Three resolution steps were used ($64 \times 64 \times 40$, $128 \times 128 \times 80$, and $256 \times 256 \times 160$) and the regularization parameter α was set to 150, 35 and 6 for the three resolution steps for the warp using LS as well as CC, and to 0.275, 0.066 and 0.011 for the warp using MI. These values were chosen so that the average variation of the regularization term in Equation (10) achieves the same order of magnitude as the average variation of the similarity term during the warp iterations. The filtering parameter β which controls the Gaussian FWHM (measured in vector indices) of the direct smoothing of the vector field between iterations chosen to 0.65, 0.41 and 0.38 for the three resolution steps, or 2.60, 1.64 and 1.52 image voxels, respectively.

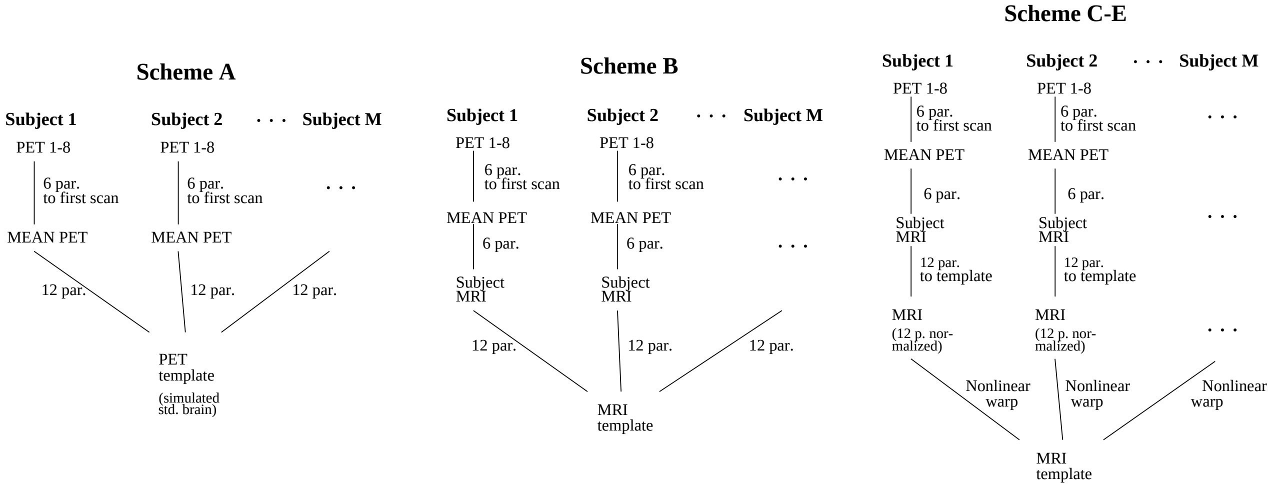


Fig. 2. The three alignment schemes used for co-registering PET images across subjects.

IV. Data Analysis

A. Multivariate Functional Analysis

Schemes A, B and the three different non-linear registrations for Scheme C produced five sets of stereo-tactically registered PET scans. These scans of each data set were passed through an SSM analysis [33], [34] and a subsequent Canonical Variate Analysis (CVA, see e.g. [36, Ch. 12]) of the resulting eigenvectors.

The SSM model is based on an orthogonal decomposition of the covariance structure of the log-transformed data, with a global scaling factor g_q for each scan. Let v_{iq} denote the detected $\left[{}^{15}\mathrm{O}\right]$water concentration at each voxel location i in scan q. With this notation, the SSM model is

$$v_{iq} = g_q\left(i_{iq} + r_i\right), \qquad \sum_i i_{iq} = 0 \tag{14}$$

$$i_{iq} = \sum_k^K h_{ik}\sqrt{\lambda_k}\, s_{kq} = \sum_k^K h_{ik}\mathrm{SSF}_{\mathrm{qk}}, \tag{15}$$

where K is the number of orthogonal components in the model. In the analysis used in the following, K equals the estimated number of degrees of freedom in the data matrix. The vector r_i is the mean activation image, and the i_{iq} matrix contains the interaction terms. This interaction matrix is decomposed using the orthonormal vectors h_k each of which describe a covariance relationship between the voxels. The eigen-vectors SSF_{qk} have corresponding eigenvalues λ_k, $\lambda_1 > \lambda_2 > \ldots > \lambda_K$, which describe the amount of variance accounted for by the k'th eigen-vector[3].

The SSM analysis is purely data-driven, which means that no information about the particular activation (paradigm) state goes into the model. Nevertheless, in multisubject activation experiments it is commonly the case that the first M-1 eigenvectors (M equals the number of subjects) describe variation between individuals and the following eigenvectors describe variation related to the activation experiment, see e.g. [38], [39]. This division of signal variance into an inter- and an intra-subject component is also visible in Figure 6 from the drop in the magnitude between eigenvalues 4 and 5 and was also found in the original group of 8 subject's PET scans [40].

For the particular experiment there were 6 different saccadic frequencies and 1 fixation condition. The variation between these 7 groups may be analyzed within the space spanned by the SSM eigenvectors using a Canonical Variate Analysis (CVA) [36, Ch. 12] to find a low dimensional subspace containing all the experimental structure as previously utilized in [39], [41]. The CVA analysis generates an orthogonal decomposition of the between group variance, so that the n'th canonical vector holds the n'th largest ratio of between group covariance and within group covariance. The CVA is performed only in the subspace spanned by the 12 largest SSF eigenvectors. The eigenvalues of the CVA analysis describe the distribution of the total variance into the respective canonical vectors. The computational details are outlined the Appendix.

A.1 Univariate Variances

More direct univariate variance measures were also computed. Let scan index $q = 1 \ldots Q$ be divided into subject $m = 1 \ldots M$ and session scan number $l = 1 \ldots L$. The total inter- and intra-subject variances normalized by the number of voxels retained, I, are computed from the interaction i_{iq}:

$$\sigma^2_{\mathrm{TOTAL}} = \frac{1}{I(Q-1)}\sum_{i,l,m}(i_{ilm} - i_{i\cdot\cdot})^2 = \frac{1}{I(Q-1)}\sum_{i,l,m} i^2_{ilm}$$

$$\sigma^2_{\mathrm{INTRA}} = \frac{1}{I(Q-M)}\sum_{i,l,m}(i_{ilm} - i_{i\cdot m})^2 \tag{16}$$

$$\sigma^2_{\mathrm{INTER}} = \frac{1}{N(M-1)}\sum_{i,m}(i_{i\cdot m} - i_{i\cdot\cdot})^2 = \frac{1}{I(M-1)}\sum_{i,m} i^2_{i\cdot m}$$

[3] If $i_{iq} = \sum_{k=1}^K \mathrm{SSF}_{\mathrm{qk}}\mathrm{h}_{\mathrm{ik}}$ then the matrices h_{ik} and SSF_{qk} are in essence the components of a Singular Value Decomposition (SVD, see e.g. [37] although they are not computed directly as such because of the g_q factors, see [33]) of the interaction matrix i_{iq}, in matrix notation: $i_{iq} = (\mathbf{USV}^T)_{iq}$ with $\mathbf{U}_{ik} = h_{ik}$ and $(\mathbf{SV}^T)_{qk} = \mathrm{SSF}_{qk}$.

B. Anatomical Analysis

The inter-subject variance may also be computed from the MR images. Figure 3 shows selected orthogonal slices of the template MR and the averaged registered MR scans for each of the results of 12-parameter alignment, and warp using MI and LS. The left column is the template volume, and the three other columns are the averaged aligned MR volumes using the three methods. Clearly, the warp using MI as similarity measure produces images that vary the least from the template volume, so that the resulting mean image is sharpest, followed by the warp using CC and LS.

The non-linear registrations resulted in a better alignment of the gross shape of the brain. The MI seems to have performed a slightly better alignment than CC and LS, but none of the registration methods were able to perfectly match the cerebral sulci.

C. Functional Analysis Results

Numerical results of the analysis are summarized in Table III. Note that the non-linear registration retains more voxels in the analysis, the warp using MI retained 2.5% more voxels than the 12-parameter method. This is occurs because the non-linear registration fits the gross shape of the brain quite well. We only retain voxels that are present in all scans, and this makes the analysis quite sensitive to registration errors at the edges of the masks—in this way the number of voxels becomes a measure of registration performance, and the non-linear registration is seen to best preserve the field of view of the analysis. Since most of the extra voxels in the non-linear registration analysis reside in grey matter areas of the cortex, we may be retaining important information about cortical activation that would be excluded using the 12-parameter alignment. Note that the number of voxels for Scheme A (PET-to-PET registration) is not directly comparable to the other schemes since the PET template brain was smaller than the MR template used in Schemes B and C–E. Figure 4 shows horizontal slices (+45 mm above and 10 mm below AC-PC plane) of the first canonical vector, i.e. the direction in scan space of largest functional discrimination between the groups representing saccadic frequencies. It is evident that the non-linear registration (the two right-most images) is better at preserving edge voxels in the analysis. The upper slice contains an activation focus in the dorso-lateral prefrontal cortex, which was undetected by the 12-parameter affine registration scheme B because the voxels near the edge of the brain were masked out. The major activation foci in the visual cortex were captured by all the registration schemes tested.

The change in total, inter- and intra-subject variance is shown graphically in Figure 5: The drop in total variance ($2.09/2.06 \cdot 10^{-2}$ affine A/B to $1.85/1.76/1.75 \cdot 10^{-2}$ non-linear C/D/E) is seen to arise mainly from a change in the inter-subject variance; the intra-subject variance is virtually unaffected by the alignment scheme. The inter-subject variance drop is absorbed mainly by the first 4 eigenvectors of the SSM analysis, as can be seen in Figure 6. The plot shows the SSM eigen-spectra (λ_k) with each eigenvalue normalized by the number of voxels retained in the analysis. Also, the sum of the first 4 eigenvalues listed in Table III reflects the change in inter-subject variance σ^2_{INTER}.

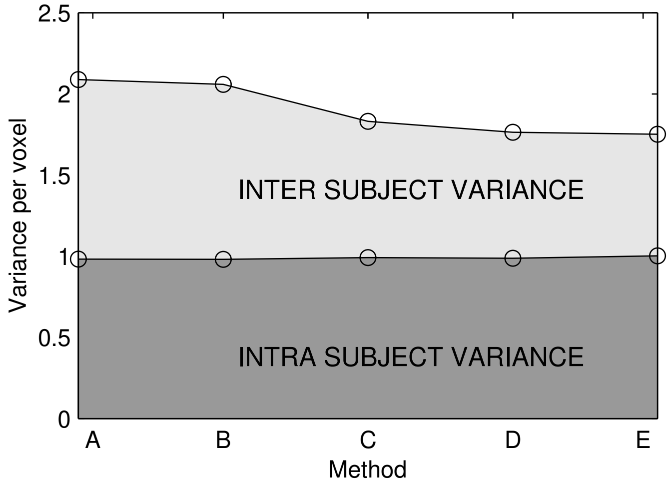


Fig. 5. Division of the total functional variance into inter- and intra subject contributions. A: Alignment using PET to PET 12-parameter affine; B: 12-parameter affine via MRI; C–E: Non-linear warp using a LS, CC and MI respectively as similarity measures.

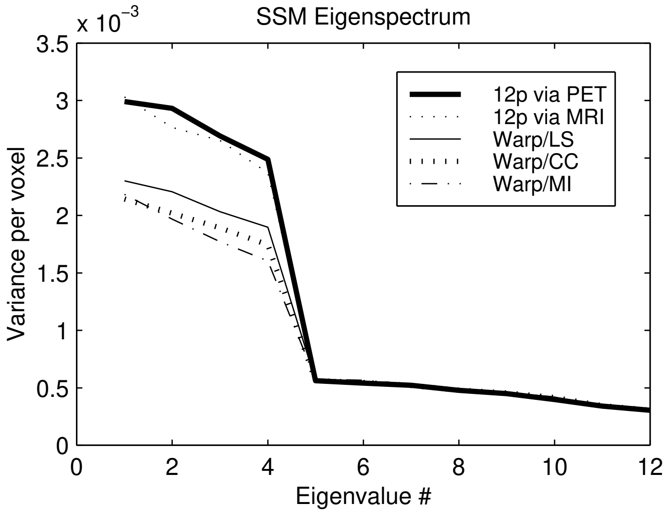


Fig. 6. The plot shows the SSM eigenspectrum obtained from each of the 2 affine and 3 non-linear alignment experiments.

These results indicate that Scheme E (warp using MI) may be the superior registration method. But until now, we have been looking only at very direct variance measures, and we have seen that Scheme E has the largest impact on the functional alignment and reduces the inter-subject variance the most. This way of evaluating the registration does not measure how good a *functional* alignment we have achieved, i.e. how good is the signal to noise ratio of the observed brain signal. In the following we shall investigate how well the brain state is reflected in the scans, assuming that the optimal alignment is the one that is able to extract the brain state most clearly. A ideal approach would be to

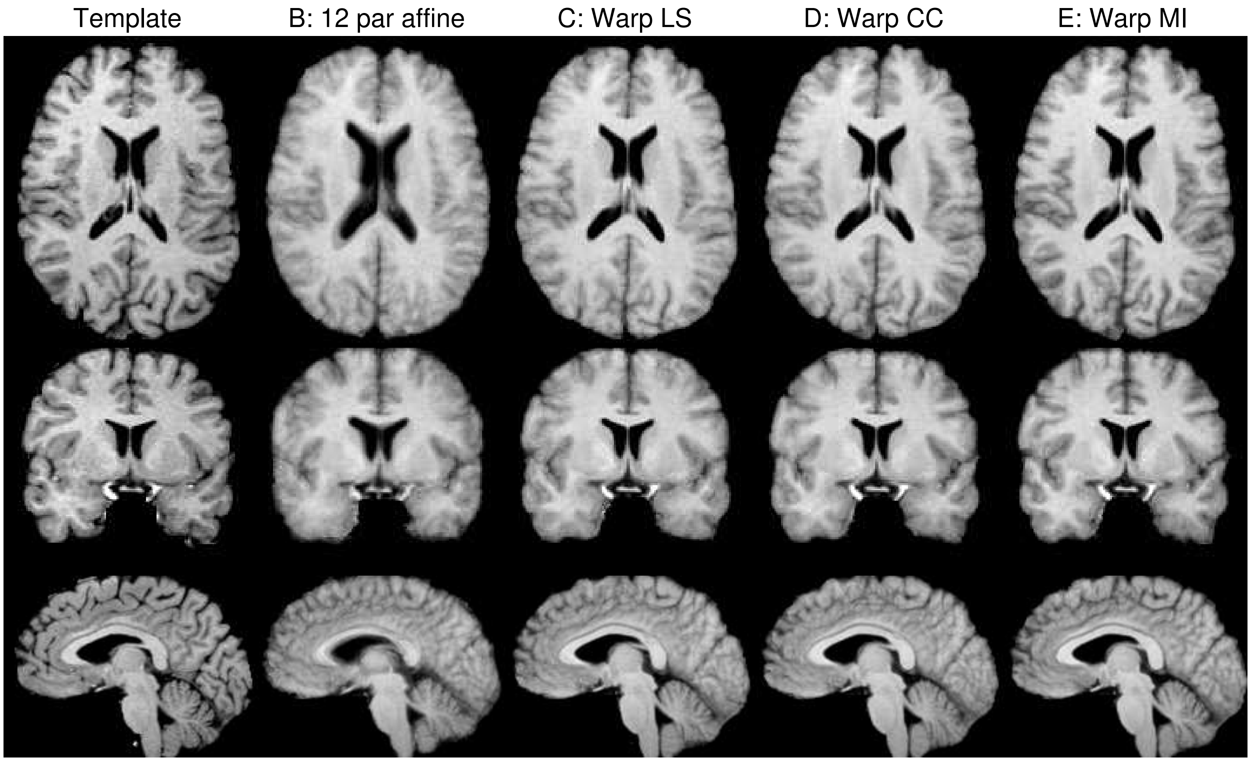


Fig. 3. Comparison of MR alignments. Leftmost image is a horizontal slice of the template volume. The four other images are mean slices across aligned subjects of the three alignment methods : 12-parameter affine and warp using the LS, CC and MI similarity measures respectively. The sharper image of MI indicates better alignment by this method.

TABLE III

Comparison of effect of the warp vs. affine on selected variance measures per voxel from the anatomical (MRI) and functional (PET) analysis.

Method	A	B	C		
	12p PET-PET	12p via MRI	C: Warp/LS	D: Warp/Corr	E: Warp/MI
[a]MRI σ_{INTER}	-	30.5	26.6	25.4	24.8
N.o. PET voxels retained, I	67621[b]	81689	83349	82908	83773
σ^2_{TOTAL}	$2.089{\cdot}10^{-2}$	$2.060{\cdot}10^{-2}$	$1.832{\cdot}10^{-2}$	$1.764{\cdot}10^{-2}$	$1.752{\cdot}10^{-2}$
σ^2_{INTER}	$1.349{\cdot}10^{-2}$	$1.315{\cdot}10^{-2}$	$1.024{\cdot}10^{-2}$	$0.946{\cdot}10^{-2}$	$0.913{\cdot}10^{-2}$
σ^2_{INTRA}	$1.094{\cdot}10^{-2}$	$1.093{\cdot}10^{-2}$	$1.105{\cdot}10^{-2}$	$1.100{\cdot}10^{-2}$	$1.117{\cdot}10^{-2}$
SSM: $\sum_{i=1}^{4} \lambda_i / N$	$1.108{\cdot}10^{-2}$	$1.082{\cdot}10^{-2}$	$0.844{\cdot}10^{-2}$	$0.780{\cdot}10^{-2}$	$0.753{\cdot}10^{-2}$
CVA: $\lambda_1 / \sum_{i=2}^{5} \lambda_i$	5.505	5.803	7.653	7.397	7.307

[a]MRI σ_{INTER} is measured per voxel inside brain mask.
[b]The number of voxels retained in the PET-PET analysis (A) is not comparable to the MRI registrations because the PET template was smaller than the MRI template used in Schemes B and C–E.

measure the reproducibility or generalization ability of the spatial activation patterns obtained, but this would require a data set with many more than 5 subjects as investigated here (e.g. [42]).

The canonical variate analysis gives information about how the frequency of the saccadadic eye movement affects the scans in a low-dimensional subspace embedded in a high-dimensional space. All scans are grouped according

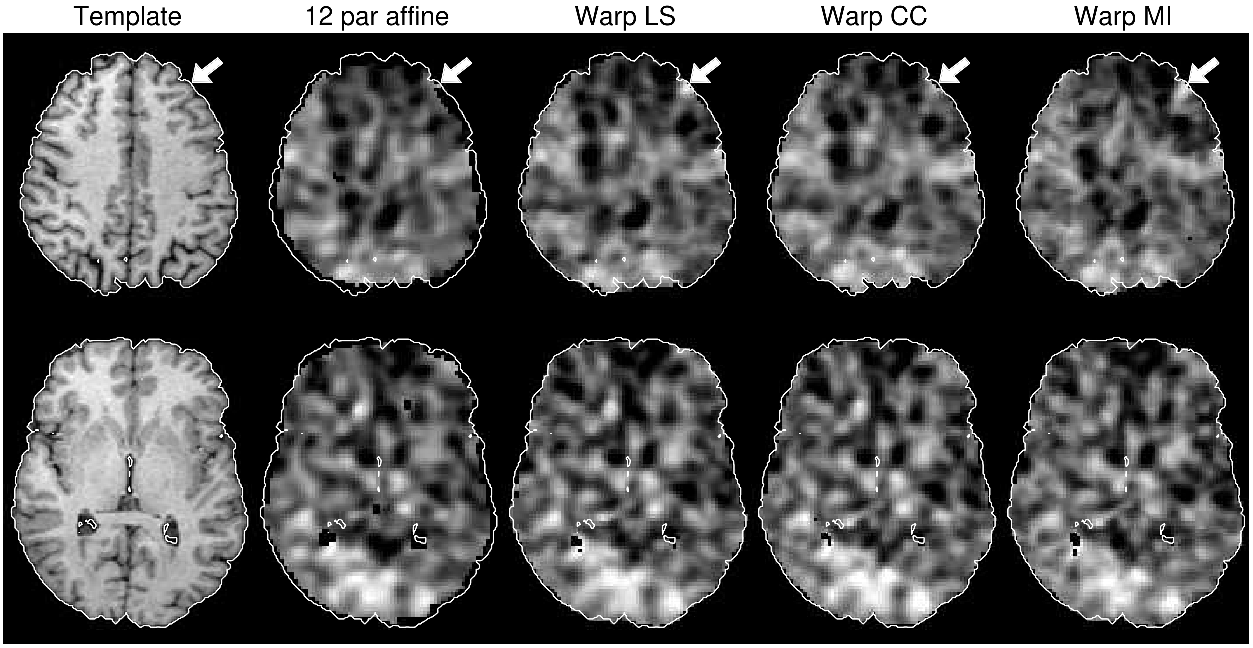


Fig. 4. Comparison of functional activation patters. Leftmost image columns shows horizontal slices (+45 mm above AC-PC plane and −10 mm below) of the template volume. The four other columns shows corresponding slices of first canonical component as obtained from the analysis of aligned subjects of the four alignment methods : 12-parameter affine and warp using the LS, CC and MI similarity measures respectively. Note that the increased number of voxels in the non-linear cases results in a larger field of view of the end analysis, causing the hot spot in the dorso-lateral prefrontal cortex to be overlooked by the affine methods.

to the saccadic eye movement frequency (with 7 groups in all). The canonical vectors form an orthogonal basis that separates these groups the most with respect to the spread within groups. If the relation between the scan and the saccadic frequency is purely linear, the variation between the 7 frequencies can be described adequately with a linear model. Figure 7 shows a plot of the canonical projections for all 40 scans registered using Scheme B. The circles indicate 90% confidence areas for the location of each saccadic frequency group mean [36, Ch. 12]. Although mild nonlinearities in the saccadic functional activation patterns from this experiment have been reported ([40],[43]), Figure 7 demonstrates that the majority of the group structure is well described by the first canonical component. We therefore assume that the remaining canonical components (λ_2–λ_6) describe 'noise' in the activation, i.e. patterns not directly related to the paradigm. Thus, the ratio of the first canonical component to the sum of the rest ("Signal to Noise Ratio") is an indication of how well the brain activation signal is captured. As shown in Table III it is actually Scheme C with a ratio of 7.7 which performs best in this sense, while Scheme E with the lower ratio of 7.3 produced the lowest inter-subject variance. We also see that the non-linear registration methods tested all outperform the 12 parameter affine registration of Schemes A and B. Figure 8 shows a scatter plot of the first versus second canonical eigenvalue from the CVA analysis for each of the 5 alignment Schemes A–E.

These results seem to indicate that it is possible to 'overwarp' the data: Instead of registering functionally equivalent areas among subjects, the more detailed registration (e.g. Scheme E) applies more active deformations which *degrade* the functional registration. To illustrate this, we compute the Jacobian of the non-linear transformations. Histograms of Jacobian values (excluding background voxels) averaged over the 5 subjects are shown in Figure 9. The transformation computed by MI involves more compression ($J < 1$) and expansion ($J > 1$) of voxel elements than the transformation generated by CC and LS.

The above discussion thus leads us to conclude that although the warp of Scheme E does the best job at reducing inter-subject variation, we seem to be better off applying a registration method which does not go into the same level of detail. Presumably this is an issue of choice of regularization parameters for the warp, rather than an issue of selecting among similarity measures, but it is our experience that the MI similarity measure offers the possibility of performing registration at a higher level of detail than is accomplished with the CC and LS measures.

V. Conclusion

We have presented a new method for performing fast non-linear registration of MR images. A lookup-table scheme has been presented and demonstrated to be an efficient tool for the implementation and testing of different voxel similarity measures using a model based on the joint probability distribution of matched voxel intensities.

The method has been tested on a functional $[^{15}O]$water PET experiment and demonstrated to produce a registration superior to a conventional 12-parameter affine registration method. One important improvement was the enhanced registration at the edge of the cortex. This effect

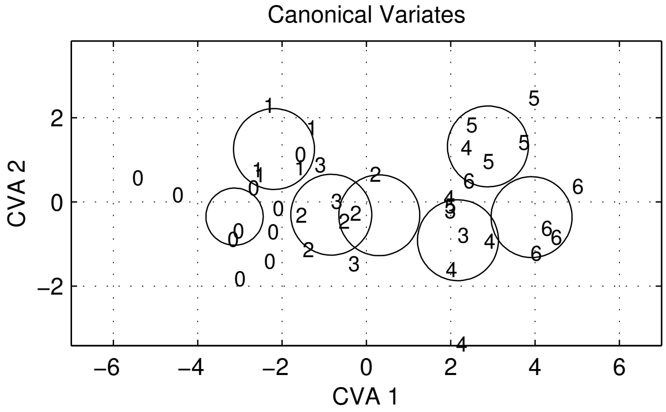


Fig. 7. Plot of first versus second canonical projections for the CVA analysis of the groupings 0: fixation, 1–6: 0.05, 0.1, 0.3, 0.5, 0.7 and 0.9 Hz saccadic eye movements. The circles are rough 90% confidence areas for the group means. The analysis was run using the 12-parameter affine registration via MRI (Scheme B).

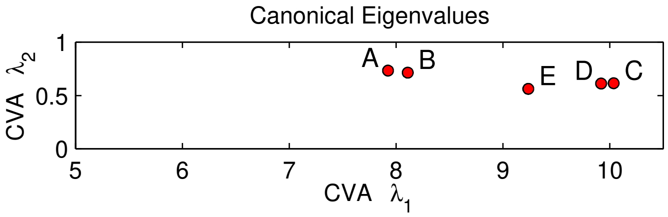


Fig. 8. Scatter plot of first versus second canonical eigenvector of a canonical analysis of the distribution of 7 functional activation states. The non-linear registration methods (C, D& E) achieve a larger 1st eigenvalue indicating better alignment. A: Alignment using PET to PET 12-parameter affine; B: 12-parameter affine via MRI; C–E: Non-linear warp using a LS, CC and MI respectively as similarity measures.

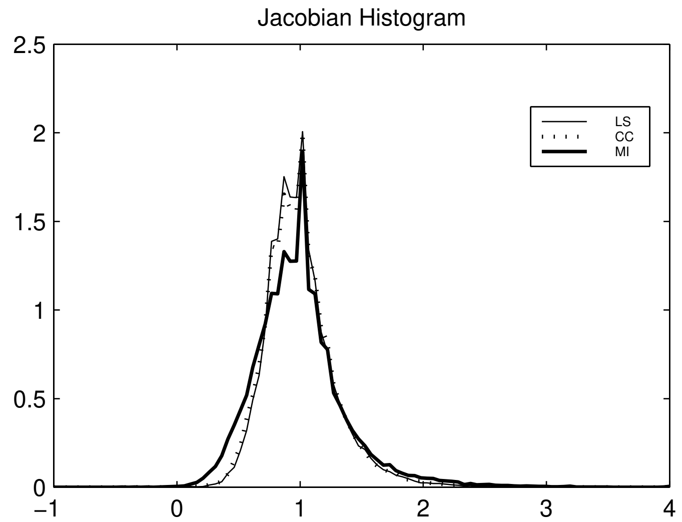


Fig. 9. Histogram of Jacobian values for the transformations computed by MI and LS. Background voxels are left out. Notice that the MI transformation has more compression ($J < 1$) and expansion ($J > 1$) of voxels than both CC and LS indicating a more active deformation.

is expected to be even more pronounced as the number of subjects in the study increases. Furthermore, a canonical analysis of the functional activation showed that noise was removed from the data: For the non-linear registration methods tested a larger estimated functional "signal-to-noise ratio" was measured: 7.65 for the warp using LS (Scheme C) compared to 5.80 for the 12-parameter normalized data (Scheme B). The experiment described here regrettably does not contain enough subjects to produce reliable error bars on these results—more than 5 subjects are needed to get a proper statistic of the inter-subject variation. Whether or not one can expect similar improvements in functional signal space by applying the registration techniques described here on other datasets presumably depends on factors such as image resolution and noise but also more importantly on the specific activation paradigm. Most of the activation hot-spots are handled quite well by the well-known 12 parameter affine registration methods and we have also demonstrated that there was no advantage in using matched MRI scans with the simpler affine registration procedure. If the paradigm in question generates activation in cortical areas that are already handled quite well by the simpler methods there may be little advantage in introducing the overhead of non-linear registration techniques.

VI. Acknowledgments

This work was supported partly by the Human Brain Project grant P20 MH57180, the Danish Research councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center (CONNECT), the SLOAN Center for Theoretical Neurobiology at the SALK institute CA, and Knud Højgaards fond, Denmark. The authors would further like to thank Terry Sejnowski, Alan Evans, Chin-Tu Chen and Yali Amit for valuable support and discussions.

Appendix

I. Canonical Variate Analysis

The canonical variate analysis stems from the multivariate analysis of variance and may be thought of as the analogue of PCA for grouped data. Consider an experiment with J conditions (or groups) and n_j, $j = 1 \ldots J$ samples of the distribution

$$\boldsymbol{x}_{ij} \in N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}), \qquad i = 1 \ldots n_j,\ j = 1 \ldots J \tag{17}$$

so that each $\boldsymbol{x}_{ij}$ is a vector of size $p \times 1$. One may test if the $\boldsymbol{\mu}_j$'s are different, i.e. whether the hypothesis H_0 : $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = ... = \boldsymbol{\mu}_J$ can be rejected. This is done using a standard likelihood ratio test, see e.g. [36, Ch. 12]. If we are using the CVA to analyze functional brain activation in the saccadic eye movement experiment, we can group the scans according to the paradigm, in this case the saccadic frequency. Each vector $\boldsymbol{x}_{ij}$ represents[4] a brain scan with

[4]Traditionally, the $\boldsymbol{x}_{ij}$ would represent an entire scan, but this causes problems in the canonical analysis since the within group covariance matrix $\mathbf{W}$ (which is a $p \times p$ matrix) becomes singular when $p > Q - J - 1$, where is the total n.o. scans. A way to circumvent

LED presentation frequency indexed by j and subject and repetition index i. Usually H_0 will be rejected (unless the paradigm did not induce significant brain activation) so we may ask more specifically about the dimensionality of the space spanned by the $\boldsymbol{\mu}$'s. Note that this is different than the univariate ANOVA case where the dimensionality is either 0 or 1. The canonical analysis performs an orthogonal decomposition of this space, so that we can inspect and test the dimensionality. Let

$$\begin{aligned} \mathbf{W} &= \sum_{i,j}(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}}_j)^T, \\ \mathbf{T} &= \sum_{i,j}(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}})(\boldsymbol{x}_{ij} - \bar{\boldsymbol{x}})^T, \text{ and} \\ \mathbf{B} &= \sum_{i,j} n_j(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_j - \bar{\boldsymbol{x}})^T \end{aligned} \tag{18}$$

where $\bar{\boldsymbol{x}}_j$ is the mean of the group that scan number j belongs to and $\bar{\boldsymbol{x}}$ is the global mean, so that $\mathbf{T} = \mathbf{W} + \mathbf{B}$. The canonical analysis proceeds as the eigenvector analysis of the matrix $\mathbf{W}^{-1}\mathbf{B}$. The r'th canonical vector is the r'th eigenvector of this matrix and describes the linear function which obtains the largest separation of the J groups under the condition that the vector is orthogonal to the $r-1$ previous canonical directions. Note that the first canonical variate is equivalent to the Fisher's linear discriminant for optimal linear discrimination between groups. The CVA analysis can be understood as the principal components analysis of the between group covariance matrix $\mathbf{B}$ in a coordinate system transformed so that the within group covariance is unity. A scatter plot of the 1st versus 2nd canonical coordinates as shown in Figure 7 can give a useful picture of the structure in the data. In such a plot, the axes are in units of standard deviations of a single data point, and we get an indication of the strength of the separation of the individual groups.

References

[1] S. C. Strother, J. R. Anderson, X. L. Xu XL, J. S. Liow, D. C. Bonar, and D. A. Rottenberg, "Quantitative comparisons of image registration techniques based on high-resolution mri of the brain.," *Journal of Computer Assisted Tomograpy*, vol. 18, pp. 954–962, 1994.

[2] R. P. Woods and S. R. Cherry, "Mri-pet registration with automated algorithm.," *Journal of Computer Assisted Tomography*, vol. 17(4), pp. 536–546, 1993.

[3] R. P. Woods, S. R. Cherry, and J. C. Mazziotta, "Rapid automated algorithm for aligning and reslicing pet images.," *Journal of Computer Assisted Tomography*, vol. 16, pp. 620–633, 1992.

[4] R. P. Woods, Scott T. Grafton, Colin J. Holmes, Simon R. Cherry, and John C. Mazziotta, "Automated image registration: I. general methods and intrasubject, intramodality validation," *Journal of Computer Assisted Tomography*, 1997, in press.

[5] R. P. Woods, Scott T. Grafton, John D.G. Watson, Nancy L. Sicotte, and John C. Mazziotta, "Intersubject validation of linear and nonlinear models," *Journal of Computer Assisted Tomography*, 1997, in press.

this problem is to use the p largest SSM eigenvectors to represent the scans in a lower dimension. In other words, x_{ij} represents the p values of the i'th scan in the j'th group for the p SSM eigenvectors with the largest eigenvalues. The dimensionality p should be chosen so that i) it is large enough to embed the expected variation introduced by the paradigm ii) smaller than $Q - J - 1$. In the experiment of saccadic eye movements described in this paper, p was 12.

[6] J. Talairach and P. Tournoux, *Co-planar Stereotactic Atlas of the Human Brain: 3-Dimensional Proportional System. An approach to Cerebral Imaging*, Georg Thieme Verlag, 1988.

[7] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall, 1982.

[8] Y. Kosugi, M. Sase, H. Kuwatani, N. Kinoshita, T. Momose, J. Nishikawa, and T. Watanabe, "Neural network mapping for nonlinear stereotactic normalization of brain mr images," *Journal of Computer Assisted Tomography*, vol. 17(3), pp. 455–460, 1993.

[9] D. L. Collins, P. Neelin, T. M Peters, and A. C. Evans, "Automatic 3d intersubject registration of mr volumetric data in standardized talairach space," *Journal of Computer Assisted Tomography*, vol. 18(2), pp. 192–205, 1994.

[10] D. L. Collins, T. M Peters, and A. C. Evans, "An automated 3d non-linear image deformation procedure for determination of gross morphometric variability in human brain," *Proceedings of Conference on Visualization in Biomedical Computing*, vol. SPIE Vol. 2359, pp. 180–190, 1994.

[11] M. E. Alexander and R. L. Somorjai, "The registration of mr images using multiscale robust methods," *Magnetic Resonance Imaging*, vol. 14, no. 5, pp. 453–468, 1996.

[12] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision and Graphics Image Processing*, vol. 46, pp. 1–21, 1989.

[13] A. C. Evans, D. L. Collins, and C. J. Holmes, "Automatic 3d regional mri segmentation and statistical probability anatomy maps," *In press*, 1995.

[14] U. Grenander, M. I. Miller, and G. E. Christensen, "Deformable anatomical data bases using global shape models," in *1992 EIHB Workshop*, 1992.

[15] S. Minoshima, R. A. Koeppe, K. A. Frey, and D. E. Kuhl, "Anatomic standardization - linear scaling and nonlinear warping of functional brain images," *Journal of Nuclear Medicine*, vol. 35(9), pp. 1528–1537, Sept. 1994.

[16] Y. Amit, "A non-linear variational problem for image matching," *Siam Journal of Scientific Computation*, vol. 15, no. 1, January 1994.

[17] J. Boes, B. Kim, K. A. Frey, and C. Meyer, "Mri-pet head registration via iterative minimization of mutual information," in *Proceedings of the 2nd Int.Conf. on Functional Mapping of the Human Brain*. June 1997, p. 149, Academic Press.

[18] B. Kim, J. Boes, K. A. Frey, and C. Meyer, "Mutual information for automated unwarping of rat brain autoradiographs," *Neuroimage*, vol. 5(1), pp. 31–40, Jan. 1997.

[19] C. Studholme, D. L. G. Hill, J. Wong anf M. N. Maisey, and D. J. Hawkes, "Registration measures for automated 3d alignment of pet and intensity distorted mr images," in *Proceedings in Image Fusion and Shape Variability Techniques*. July 1996, pp. 186–193, IEEE Computer Society Press ISBN 0-8186-7367-2.

[20] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Incorporating connected region labelling into automated image registration using mutual information," in *Proceedings of the IEEE workshop on Mathematical Methods in Biomedical Image Analysis, San Fransisco CA*. June 1996, pp. 23–31, IEEE Computer Society Press ISBN 0-8186-7367-2.

[21] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Automated three-dimensional registration of magnetic resonance and positiron emission tomography brain images by multiresolution optimisation of voxel similarity measures," *Medical Physics*, vol. 24, pp. 24–35, 1997.

[22] S. Geeman and D. Geeman, "Stochastic relaxation, gibbs distributions, and the baysian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Pattern Analysis and Machine Intelligence, Nov. 1984.

[23] S. Lakshmanan and H. Derin, "Gaussian markov random fields at multiple resolutions," *Markov Random Fields, Theory and Application*, pp. 131–157, 1993.

[24] G. E. Christensen, M. I. Miller, and M. Vannier, "A deformable magnetic resonance textbook bason on elasticity," in *Application of Computer Vision in Medical Image Processing*. March 1994, Stanford University.

[25] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "A deformable neuroanatomy based on viscous fluid mechanics," in *Proc. of the 1993 Conference on Information Sciences and Systems*, Mar. 1993, pp. 211–216.

[26] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "3d

brain mapping using a deformable neuroanatomy," *Physics in Medicine and Biology*, vol. 39, pp. 609–618, 1994.

[27] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "Deformable templates using large deformation kinematics," *IEEE Transactions of Image Processing*, vol. 5(10), pp. 1435–1447, 1996.

[28] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," in *Proc. of the National Academy of Science*, December 1993, vol. 90:24, pp. 11944–11948.

[29] K. Friston, J. Ashburner, C. D. Frith, J-B. Poline, J. D. Heather, and R. S. J. Frackowiak, "Spatial registration and normalization of images," *Human Brain Mapping*, vol. 2, pp. 165–189, 1995.

[30] D. L. Collins, G. Le Goualher, R. Venugopal, A. Caramanos, A. C. Evans, and C. Barillot, "Cortical constraints for non-linear cortical registration," *Lecture Notes in Computer Science, Visualization in Biomedical Comptuting*, pp. 307–316, 1996.

[31] I. Law, C. Svarer, and O. B. Paulson, "Characterization of the cortical responses during the performance of reflexive and antisaccadic eye movements," *Human Brain Mapping*, p. 1:323, 1995.

[32] T. R. DeGrado, T. G. Turkington, J. J. Williams, C. W. Stearns, J. M. Hoffman, and R. E. Coleman, "Performance characteristics of a whole-body pet scanner," *Journal of Nuclear Medicine*, vol. 35, pp. 1398–406, 1994.

[33] J. R Moeller and S. C. Strother, "A regional covariance approach to the analysis of functional patterns in positron emission tomographic data," *Journal of Cerebral Blod Flow Metabolism*, vol. 11, pp. A125–135, 1991.

[34] J. R Moeller, S. C. Strother, J. J. Sidtis, and D. A. Rottenberg, "Scaled subprofile model: A statistical approach to the analysis of functional activation patterns in positron emmision tomographic data," *Journal of Cerebral Blod Flow Metabolism*, vol. 7(5), pp. 649–658, 1987.

[35] S. C. Strother, J. R. Anderson, K. A. Schaper, J. J. Sidtis, J. S. Liow, R. P. Woods, and D. A. Rottenberg, "Principal component analysis and the scaled subprofile model compared to intersubject averaging and statistical parametric mapping: I. "Functional connectivity" of the human motor system studied with [15-o]water pet," *Journal of Cerebral Blood Flow and Metabolism*, vol. 15, pp. 738–753, 1995.

[36] K. V. Mardia and J. T. Kent, *Multivariate Analysis*, Academic Press, 1979.

[37] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[38] S. C. Strother, J. R. Anderson, J. J. Sidtis, K. A. Schaper, L. K. Hansen, J. S. Liow, and D. A. Rottenberg, "Functional image registration metrics based on orthogonal subspaces," *Mapping and Modeling of the Human Brain*, 1997.

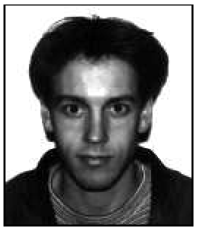
[39] S. C. Strother, J. R. Anderson, K. A. Schaper, J. S. Sidtis, and D. A. Rottenberg, "Linear models of orthogonal subspaces and activated networks from functional activation pet studies of the human brain," in *Computational Imaging and Vision, Information Processing in Medical Imaging*. 1995, vol. 3, pp. 299–310, Kluwer Academic Pubs.

[40] B. Lautrup, L. K. Hansen, I. Law, N. Mørch, C. Svarer, and S. Strother, "Massive weight sharing: A cure for extremely ill-posed problems," in *Proceedings of Workshop on Supercomputing in Brain Research: From Tomography to Neural Networks: From tomography to neural networks, HLRZ, KFA Jülich, Germany*, H. J. Hermann, D. E. Wolf, and E. P. Pöppel, Eds. Nov. 1995, pp. 137–148, World Scientific.

[41] S. C. Strother, N. Lange, R. L. Savoy, J. R. Anderson, J. J. Sidtis, L. K. Hansen, P. A. Bandettini, K. O'Craven, M. Rezza, B. R. Rosen, and D. A. Rottenberg, "Multidimensional state-spaces for fmri and pet activation studies," in *Neuroimage 3(Part 2) 2nd Int. Conf. Functional Mapping of the Human Brain*, June 1996, p. S98.

[42] S. C. Strother, N. Lange, J. R. Anderson, K. A. Schaper, K. Rehm, L. K. Hansen, and D. A. Rottenberg, "Activation pattern reproducibility: Measuring the effects of group size and data analysis models," *Human Brain Mapping*, vol. 5, pp. 312–316, 1997.

[43] N. Mørch, L. K. Hansen, S. C. Strother, C. Svarer, D. A. Rottenberg, B. Lautrup, R. Savoy, and O. B. Paulson, "Nonlinear versus linear models in functional neuroimaging: Learning curves and generalization crossover," in *Lecture Notes in Computer Science 1230: Information Processing in Medical Imaging*, J. Duncan and G. Gindi, Eds. 1997, pp. 259–270, Springer-Verlag.



Ulrik Kjems born March 1971, he obtained his Master of Science in Electrical Engineering at the Technical University of Denmark February 1995. Since then he has been working with methods for incorporating anatomical prior information into functional brain images at the Section for Digital signal Processing at the Technical University of Denmark. In addition to non-linear warping, his research interests are parameter estimation in Markov random fields and (non-)linear modelling of functional activation patterns in the human brain. http://eivind.imm.dtu.dk/staff/kjems

A Nonlinear 3D MRI Brain Co-registration Method

Ulrik Kjems† Peter A. Philipsen†
Lars Kai Hansen† Chin-Tu Chen♮
Jon Anderson‡

† Department of Mathematical Modelling
Building 349
Technical University of Denmark
DK-2800 Lyngby, Denmark

♮ Frank Memorial Image Center
Department of Radiology
University of Chicago

E-Mail : kjems@ei.dtu.dk
WWW : www.ei.dtu.dk/staff/kjems/kjems.html

February 1, 1996

Abstract

This paper describes a method for computing a 3 dimensional displacement field to align volumetric data. The method is designed for co-registering MR images of the human brain, and is based on local block correlation optimization scheme. An example using real anatomic data is presented, and different evaluation schemes are discussed.
Keywords: Non-linear warping, stereo-tactic normalization, elastic mapping, deformable template models, anatomical textbooks, inter-subject variation.

1 Introduction

In the field of computerized medical imaging, a classical task is automated interpretation of anatomical images. In this paper, this interpretation is expressed as a deformation of a template image to match an observed image. Images are volumetric, i.e. consisting of stacks of anatomical slices, recorded with uniform spacing between slices. Given two such volumetric images, the desire is to apply a deformation to the one, making it look like the other image. Any higher level information about the first image (our *template*) can thus be transformed through the deformation and projected on to the observed image. The deformation represents a *correspondence map*, which describes which voxel[1] in the template image that any given voxel in the observed image corresponds to.

Clearly, two factors should be taken into account when searching for a deformation field. One is that we have prior knowledge that the deformation field in some sense should be smooth and coherent. Secondly, for some image parts there may not even exist a corresponding area in template, due to normal anatomical variation. If one is not careful enough, a deformation field may 'invent' anatomical features which are not present in the template—anatomical details may be created by simply dragging existing tissue to create new details.

[1] A voxel is a three dimensional pixel, i.e. smallest unit in a sampled volume image.

Similarly, in the reverse case anatomical details can be removed in the deformation process. Whether or not these kinds of changes in anatomy are desired depends on the specific application.

The purpose of this work has been to develop a method for co-registering of 3D MR images of the human brain. The deformation field computed on basis of the anatomical images is transfered to functional image modalities like PET and fMRI, in order to compensate for the variation of anatomy. This implies that the correspondence map (ideally) should be understood as the regions of the brain that are believed to have the same *function*, as far as can be determined from anatomy.

In PET activation studies, the injection of the radioactive tracer is a health risk. It is therefore limited how many times a subject may be scanned—therefore several individuals are scanned in functional activation studies. A typical study could involve 8 subjects, each scanned 8 times. To combine the measurements from different subjects, there is clearly a need for some registration method, which can reduce the *inter-subject variance* in the PET images.

2 Specifying the Problem

The template volume image $\mathcal{A}$ and the data image $\mathcal{B}$ are sampled on a grid $\mathcal{V}$, according to the resolution (I, J, K) voxels. Each image may be available in M channels $\mathcal{A}^{1..M}$, $\mathcal{B}^{1..M}$. We seek a displacement field expressed as the vector field $\Phi(\boldsymbol{x})$ specifying a three dimensional displacement vector for each voxel in $\mathcal{A}$, so that the distance measure

$$\sum_{m=1}^{M} \sum_{\boldsymbol{x} \in \mathcal{V}} \Big(\mathcal{A}^m(x + \Phi(x)) - \mathcal{B}^m(x) \Big)^2 \tag{1}$$

is minimized. This paper describes an automated method for finding $\Phi(\boldsymbol{x})$ given the images $\mathcal{A}^m$ and $\mathcal{B}^m$.

2.1 Constraints and Considerations

Clearly, some constraint must be put on the solution of the displacement field. If the displacement field $\Phi(\boldsymbol{x})$ was allowed to be chosen arbitrarily, the distance measure (1) could be brought close to 0: Simply let each voxel site register with a voxel of similar intensity anywhere else in the image. This would result in an extremely 'noisy' or jiggered vector field, and would be of no particular interest.

We need to restrain the solution of $\Phi(\boldsymbol{x})$. A smoothness constraint on $\Phi(\boldsymbol{x})$ can be included in the cost function by the penalty or restriction term $R(\Phi)$

$$\sum_{m=1}^{M} \sum_{\boldsymbol{x} \in \mathcal{V}} \Big(\mathcal{A}^m(x + \Phi(x)) - \mathcal{B}^m(x) \Big)^2 + R(\Phi). \tag{2}$$

Assume that the images $\mathcal{A}^m$, $\mathcal{B}^m$ and the field Φ are incidents of stochastic variables, the posterior distribution of the displacement field may be expressed using Bayes' relation

$$\begin{array}{rcl} \underbrace{P(\Phi|\mathcal{A},\mathcal{B}))}_{\text{To be optimized}} \cdot P(\mathcal{A},\mathcal{B}) & = & \underbrace{P(\mathcal{A},\mathcal{B}|\Phi)}_{\text{Model of residual}} \cdot \underbrace{P(\Phi)}_{\text{Prior on }\Phi} \\ & \Downarrow & \\ P(\Phi|\mathcal{A},\mathcal{B})) & \propto & P(\mathcal{A},\mathcal{B}|\Phi) \cdot P(\Phi) \end{array}$$

Assuming that the residual image $\mathcal{A}(x) - \mathcal{B}\big(x + \Phi(x)\big)$ is identically distributed independent

Gaussian noise, the noise model becomes

$$P(\mathcal{A}\&\mathcal{B}|\Phi) = \frac{1}{(2\pi\sigma^2)^{IJK/2}} \exp\left[\frac{1}{2\sigma^2} \sum_{\substack{\boldsymbol{x}\in\mathcal{V}\\ m=1..M}} \Big(\mathcal{A}^m(x+\Phi(x)) - \mathcal{B}^m(x)\Big)^2\right] \tag{3}$$

where σ is a noise parameter. Taking the -log of (3) and inserting (3), the cost function (2) appears.

The prior term function determines the nature of the displacement fields, that will be generated from optimizing the cost function. In this paper, a Laplace prior model is used. In [3, 4] mechano-elastic and viscous prior models are described. Common to these models is that they assume the observed image to be a deformed instance of the template image, with additive Gaussian noise. When dealing with stereo-tactic normalization of human brain anatomies, normal variation in anatomy makes this assumption questionable. What makes

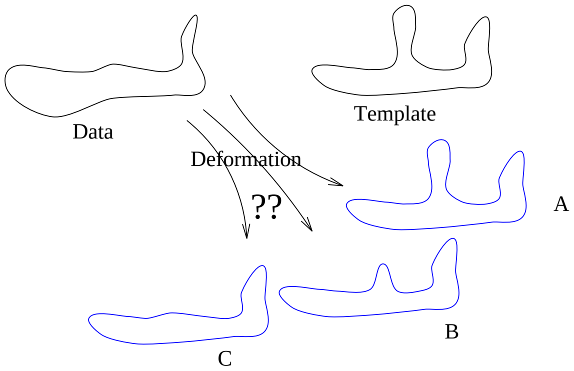


Figure 1: Warping amoeba's : Should the missing 'leg' of the data image be created by dragging out tissue from the body (a), or should the absent leg be left as it is (c)?

the topic of anatomical warping relevant is the set of common characteristics in the images. The problem is what to do with real differences in anatomy, as illustrated in figure 1.

The optimal solution would depend on the specific use: In our case the purpose of the warping is to perform stereo-tactic normalization of brain volumes from anatomical images (MR scans), and to transfer this information to images from functional imaging modalities (PET / fMRI). For this type of problems, it is not desirable to 'invent' new anatomical features by exaggerated deformations. We want to register regions, that are assumed to have the same function, without registering areas 'at any price' (meaning a desire to get a seemingly perfect match of anatomy) using deformations, that cannot be interpreted.

Only with a vast anatomical database including a full probabilistic description of anatomical variability the above problem can be solved. Such an approach, however, is orders of magnitude more complex than the deformable template technique. With templates, there is a certain limit to the amount of detail that can be registered. In other words, there is a build-in limitation to this approach.

3 The 3D Block Search Algorithm

The registration method is an iterative, coarse-to-fine direct search method, and is in fact a 3D extension of the 2D method described in [5].

Before commencing the search for the field Φ, a set of subsampled images of both the template and the data image is created, e.g. if the raw images are available in $256\times256\times64$ voxels, $32\times32\times32$, $64\times64\times64$ and $128\times128\times64$ versions are computed. Iterations proceed in two levels: At the outer level, the resolution of the images increases, transferring the current deformation field estimate to higher resolution at each iteration step. At the inner loop the deformation estimate is improved using a local search scheme.

This search proceeds as follows: The template image is divided in $P \times Q \times R$ non-overlapping blocks of voxels. The displacement field estimate Φ is represented with one vector for each of the blocks.

Block index	p, q, r
Block dimension	P, Q, R
Image index	i, j, k
Image dimension	I, J, K
Current displacement field estimate (n'th inner iteration)	$\Phi^n_{pqr},\ {\scriptstyle 0 \leq p < P,\ 0 \leq q < Q,\ 0 \leq r < R}$

For each block, do a local correlation search, computing correlation scores by shifting all voxels in a block an integer amount of indices in a neighborhood around the current displacement estimate for that block. The specific neighborhood size S is typically set to 2–3 voxels.

$$\Phi^{n+1}_{pqr} = \underset{\boldsymbol{d} \in \mathcal{R}_{pqr}}{\text{maxarg}} \left[C_{pqr}\left(\Phi^n_{pqr} + \boldsymbol{d}\right) \right] \tag{4}$$

$$\mathcal{R}_{pqr} = \{(i,j,k) | -S \leq i \leq S, -S \leq j \leq S, -S \leq k \leq S\} \tag{5}$$

$$C_{pqr}(\boldsymbol{d}) = \sum_{\substack{x_0 - P/2 \leq i < x_0 + P/2 \\ y_0 - Q/2 \leq j < y_0 + Q/2 \\ z_0 - R/2 \leq k < z_0 + R/2}} \left[\mathcal{A}\big((i,j,k) + \boldsymbol{d}\big) - \mathcal{B}\big((i,j,k)\big)\right]^2 + R_{pqr}(\Phi^n) \tag{6}$$

The last term in the cost function equation (6) is the regularizing or smoothness term. A penalty for neighborhood differences of vectors is implemented corresponding to the Laplacian

$$\left(\frac{d\Phi}{dx}\right)^2 + \left(\frac{d\Phi}{dy}\right)^2 + \left(\frac{d\Phi}{dz}\right)^2$$

For the discrete Φ this becomes

$$R_{pqr}(\Phi) = \alpha PQR \sum_{j \in \mathcal{N}_{pqr}} |\Phi_{pqr} - \Phi_{j,pqr}|^2$$

where α is the smoothness parameter and $\mathcal{N}_{pqr}$ is the 6 nearest neighbors of block pqr. The scaling PQR is included to make α independent of the block dimensions. After all the vectors in Φ have been updated, the field is smoothened by *brut de force* low pass-filtering of the vector field components. The kernel is a $3 \times 3 \times 3$ box filter, with the center value β varying to act as a smoothness control parameter. The array is normalized to sum to unity.

$$\underset{z=1}{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}} \qquad \underset{z=2}{\begin{matrix} 1 & 1 & 1 \\ 1 & \beta & 1 \\ 1 & 1 & 1 \end{matrix}} \qquad \underset{z=3}{\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}}$$

The smoothened field is then reiterated in a new block search until the field estimate no longer changes. At this point, the search proceeds to higher resolution: A new block size is selected, and the vector field estimate is resampled using bilinear interpolation to match the new image and block dimensions. The warped data image is constructed at the very end, by linear interpolation in the displacement of the blocks to get a displacement vector for each voxel in the template image.

So far, a number of search parameters have been introduced. By experience, a certain set of parameters will give good results when matching human brain anatomies—an example of a set of working parameters is listed in table 1. The parameters include the image resolutions

Outer It.	Resolution	Block Size	α	β	S	N.o. it.
1	$32 \times 32 \times 32$	$2 \times 2 \times 2$	0.0	2.0	3	4
2	$32 \times 32 \times 32$	$2 \times 2 \times 2$	50.0	7.0	2	4
3	$64 \times 64 \times 64$	$4 \times 4 \times 4$	50.0	7.0	2	2
4	$128 \times 128 \times 64$	$4 \times 4 \times 4$	50.0	7.0	2	2
5	$256 \times 256 \times 64$	$4 \times 4 \times 2$	100.0	10.0	2	2

Table 1: Typical parameters. The parameters apply for proton density weighted (20 ms echo) 8 bit MR images. Computation time for this parameter setting is around 15 min.

(I, J, K), smoothness parameters α, β, block size (P, Q, R), search region size S and the maximum number of search iterations at each resolution. The parameters are changed at each resolution in order to allow for a more efficient search for the minimum in the cost function. The performance of the algorithm depends very much on the proper setting of these parameters, which introduces the need for an evaluation scheme: We need a way of determining whether one set of parameters gives better results than another set.

3.1 Comments

In the algorithm, smoothness is enforced in a number of ways: 1) Through the Laplace neighborhood penalty, 2) through the *brut de force* filtering after each iteration, 3) by representing the field in the block resolution only, and 4) by the coarse-to-fine resolution scheme. Together, these factors make up for a fundamental weakness in the search method, namely the fact that the local correlation search tends to generate spatially high-frequent displacement fields. Slow varying displacements involving many blocks, with vector lengths greater than the selected neighborhood size S, are not found by the local searches—not without help at least. The coarse-to-fine resolution scheme helps the algorithm to overcome the range problem, while the *brut de force* filtering tends to help get past local minima in the cost function.

Some of the limitations may be overcome by choosing another search domain. In [1], the displacement field is represented by wavelet coefficients, and the search is performed by gradient descent in the wavelet coefficient domain. The smoothness constraint can then simply be imposed by limiting the number of wavelet coefficients. This approach is currently being investigated.

Computation Time

The time for computing a displacement field depends on the specific choice of parameters. For the set displayed in table 1, the computation time on a HP735 work station is approximately 15 min.

4 Evaluating Normalization Algorithms

A common problem in relation to template matching is how to evaluate the resulting displacement fields. The fundamental problem is that exists no ‘true’ solution: The anatomy in the template image cannot be matched fully to the template image. This problem comes specifically into hand, when the parameters in the cost function need to be estimated. Different schemes are suggested and used in the literature.

Anatomical landmark points (*fiducial markers*).

The displacement field is evaluated by manually placing anatomical landmarks in the template and the data images, and measuring the mismatch in the warped data landmarks and the template landmarks.

\+ Landmarks are points we trust, and know should be registered.

− Manual placement introduces factor of human error. Landmarks are very hard to place accurately ($\sim 5\,\mathrm{mm}$ std. error). It is the author's experience, that only a very limited number of landmark points exist in the human brain, that can be pinpointed accurately, and relatively few are in the cortex.

Segmentation overlap.

This method implies classifying the data and template into tissue types (e.g. white matter, gray matter, background, etc.), and then measuring the area of classification match in the images.
+ Valid if the data image is truly a deformed instance of the template.
− No test for crude deformations, i.e. testing validity of anatomical similarity of data and template.

Visual inspection of results.

The warped data image should, ideally, be similar to the template. Still, this is under the assumption, that the data image is a deformation of the template. The deformation field can be visualized in different ways, see section 5. In this way, we can check if the deformations seem 'reasonable'. + Smoothness and global features can be verified.
− Human factor should be avoided.

Apply deformation to PET and measure change in inter-subject variation.

This method is applicable in the case, where both PET and MR scans are available for each subject. The deformation field that brings the individual subject's anatomy into standard space using a common anatomical template may be applied to the functional image. Methods are known for co-registering across modalities, see f.x. [2], which is necessary in order to register the each deformation field to the functional images. The inter-subject functional variance is the variance in the functional images across subjects. If this measure is lowered after applying the normalizing deformation, we have a strong indication, that the normalization is working correctly—we are even able to compare different normalization methods quantitatively.
+ Evaluates the *functional registration.*
− Complex operation. Requires both PET and MR scans for a number of subjects.

5 Example

In this section, some preliminary results are presented. A simple landmark experiment is described, and the block search algorithm described in this paper is compared to the Wood's 12 parameter registration algorithm[2] , see [6].

Test persons were one male European (initials SCS, acting as template), and one male Asian (initials JSL, data image to be matched to template)—these were selected because Asian and European brains have relatively different anatomy. The scans consisted of 56 slices of 256×256 pixels, acquired with TE of 20 ms. Subsampled images were computed in resolutions $32 \times 32 \times 32$, $64 \times 64 \times 64$ and $128 \times 128 \times 64$ using Gaussian filtering to avoid aliasing, and spectral resampling. The parameters from table 1 were used in the block search algorithm. Computing the deformation field took approximately 30 minutes on a HP 735 work station with 128 MB of memory.

Figure 2 page 9 shows slices of the resulting images. The left column shows the data image, the warped data image is in the 2nd and 3rd columns, while the template image is to the right. The data image was warped both using the method described in this paper (3rd column) and using Wood's method (2nd column). The global shape of the brains is seen to

[2]The implementation of Wood's method was made by the group at the VA Medical Center, University of Minnesota.

be matched quite well, but details like fissures are left unmatched in both cases. The block search method seems to perform a bit better on the cortex.

Gray-codings of the displacement field components for the block search are depicted in the images in figure 3 page 10. To demonstrate another visualization approach, the deformation field is shown as a deformed mesh in figure 4 page 11. Both of these visualization approaches give the impression of a relatively smooth displacement field.

5.1 Landmark Test

A landmark test was applied for evaluating the computed displacement field. Landmarks where placed in both data and template volumes using an interactive viewer, showing saggital, axial and coronal slices, table 2 lists the 17 anatomical landmarks used in this test.

To measure the inherent error in manual landmark placement, the same landmark volume (JSL) was marked twice, by different individuals. The resulting RMSE (Root Mean Squared Error) between the two sets of landmarks was 0.4872 cm. Naturally, this error depends on the MR image quality. T1-weighted MR images were used for locating the landmarks. If the MR slices were acquired in the saggital slices in stead of axial and/or in a higher resolution, the placement of landmarks could be more accurate.

The initial difference between the landmark sets in JSL and SCS was 0.909727 cm, excluding the mean offset. This difference was lowered to 0.623141 cm after applying the deformation field found by the iterative block search algorithm. Wood's 12 parameter semi-automatic registration method was applied to the same data for comparison, it brought down the landmark mismatch to 0.648213 cm—close to the performance of the block search method.

1. Top of cerebellum.
2. Bottom of cerebellum, left.
3. Bottom of cerebellum, right.
4. Bottom of temporal lobes, left.
5. Bottom of temporal lobes, right.
6. Posterior corpus callosum.
7. Anterior corpus callosum.
8. Center of thalamus on mid-saggital section.
9. Frontal of pons.
10. Inferior colliculus.
11. Back of cerebellum, left.
12. Back of cerebellum, right.
13. Posterior horn of lateral ventricle, left.
14. Posterior horn of lateral ventricle, right.
15. Top of brain, middle line.
16. Centerline front.
17. Centerline back.

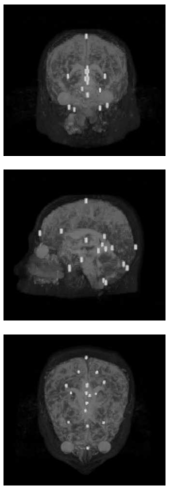


Table 2: The landmarks used for evaluating the stereo-tactic normalization.

Evaluation Using Masks

The data and template images were segmented semiautomatically into brain/non-brain regions. The masked data image was warped, and the overlap with the template was measured before and after the warp, table 3 lists the results of this test. The block search algorithm again proves to performs better than the Wood's method on this test.

6 Conclusion

A new algorithm for computing 3D displacements for stereo-tactic normalization of MR images of the human head has been presented. It has been argumented, that a proper evaluation method is needed to establish credibility in automatic registration methods. The

	Original Volumes	Warped JSL using block search	Warped JSL using Wood
Total mask volume, SCS	626077		
Total mask volume, JSL	572915	610475	648477
Difference	-53162	-15602	22400
Identical volume	545542	597616	602625
Volume of difference	107908	41320	69304

Table 3: Measuring the overlap and error volumes in segmented data and template images. The segmentation is into brain/non brain areas. Units are voxels (1 by 1 by 3 mm). The row labeled 'difference' is the difference in the total brain volumes listed in the top rows. 'Identical volume' is the total volume of overlap (i.e. voxels that are segmented as brain in both images. Likewise, 'Volume of difference' is the volume of voxels, that have different segmentation.

landmark test has shown to be rather uncertain, because the inherent uncertainty in placing anatomical landmarks by hand is too high.

The algorithm is still at the experimental state—several features need to added. F.x. a rigid rotation, scale and rotation should be included to initialize the deformation field.

The major problems with the block search method remains the setting of search parameters. In absence of a proper qualitative evaluation scheme, determining if one set of parameters is better than another is difficult.

7 Acknowledgments

Thanks to Leann Kinnunen and Jeff Yap for assistance with the anatomical landmarks. The Knud Højgaards Fond contributed with economical support.

References

[1] Yali Amit. A non-linear variational problem for image matching. *Siam J. Sci. Comput.*, 15, January 1994.

[2] Babak A. Ardekani, Michael Braun, Brian F. Hutton, Iwao Kanno, and Hidehiro Iida. A fully automatic multimodality image registration algorithm. *Journ. of Computer Ass. Tomography*, 19(4):615–623, 1995.

[3] Gary E. Christensen, R. D. Rabitt, and M. M. Miller. A deformable neuroanatomy based on viscous fluid mechanics. In *Proc. of the 1993 Conference on Information Sciences and Systems*, pages 211–216, March 1993.

[4] Gary E. Christensen, R. D. Rabitt, and M. M. Miller. 3d brain mapping using a deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.

[5] Yukio Kosugi, Mikiya Sase, Hiroshi Kuwatani, Naoyuki Kinoshita, Toshimitsu Momose, Junichi Nishikawa, and Toshiaki Watanabe. Neural network mapping for nonlinear stereotactic normalization of brain mr images. *Journal of Computer Assisted Tomography*, 1993.

[6] Woods RP, Mazziotta JC, and Cherry SR. Automated image registration. In Uemura K, Lassen NA, Jones T, and Kanno I, editors, *Proceedings of Brain PET '93 AKITA: Quantification of Brain Function.* Amsterdam, Excerpta Medica, 1993.

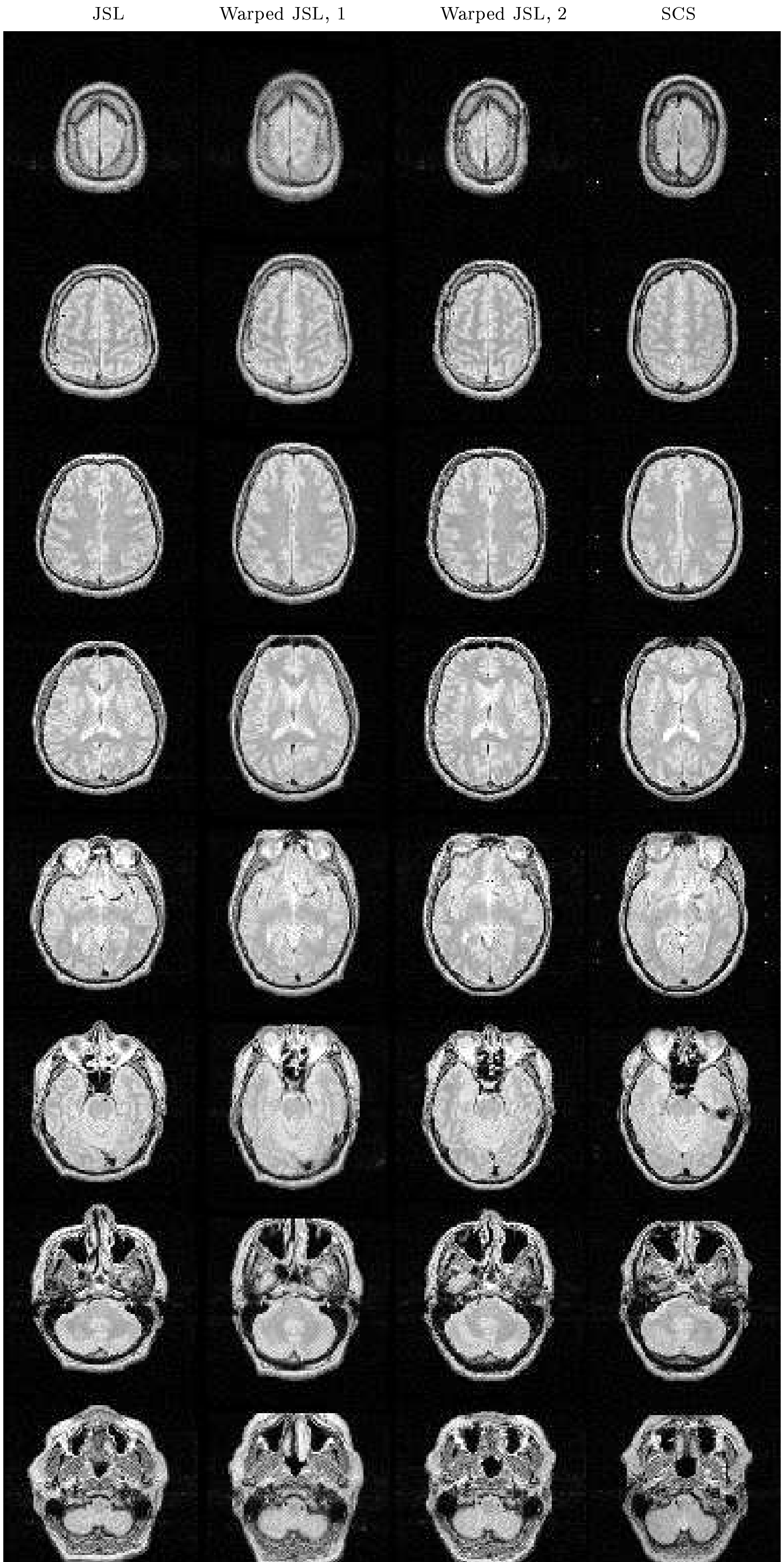


Figure 2: The leftmost column (A) shows the data image, the warped data image using Wood's method is in the 2nd column, 3rd column is the block search method described in this paper, and the template image (target) is to the right. The global shape of the brains is seen to be matched quite well, but details like fissures are left unmatched.

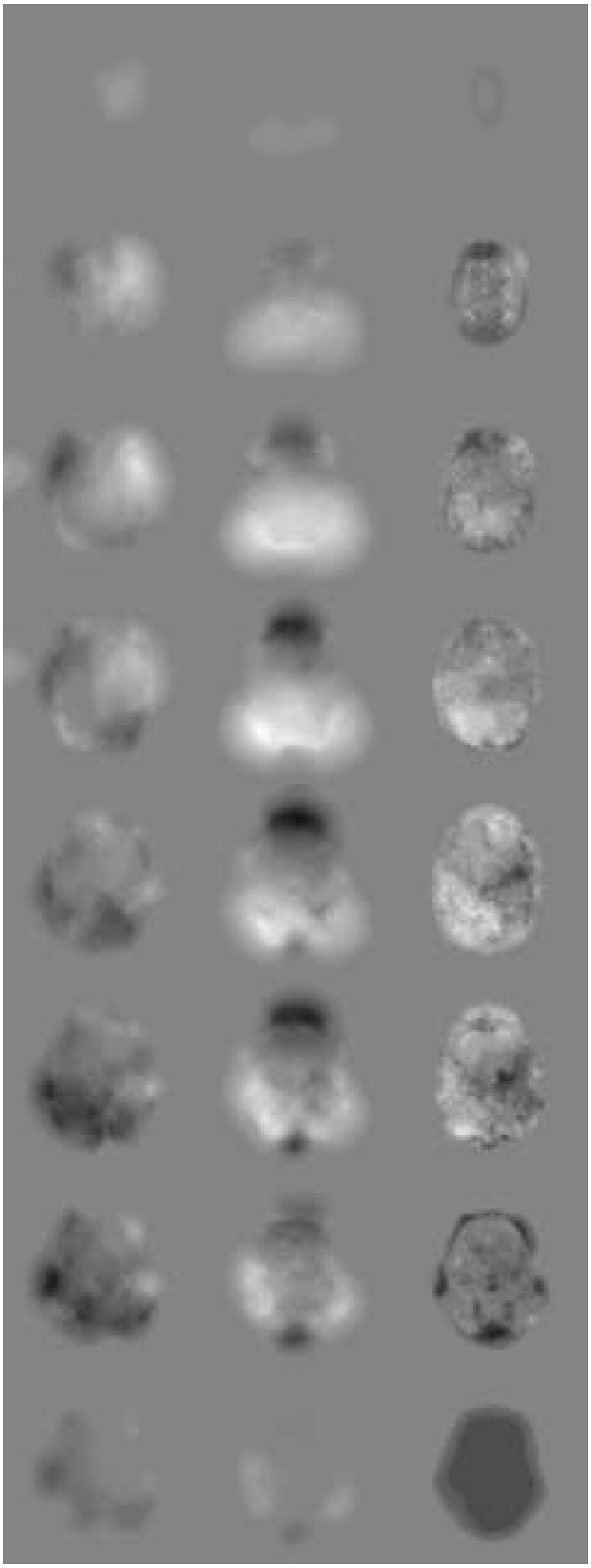


Figure 3: A grayscale coding of the displacement field. Left to right: X, Y, and Z-components of the displacement field, all in axial slices.

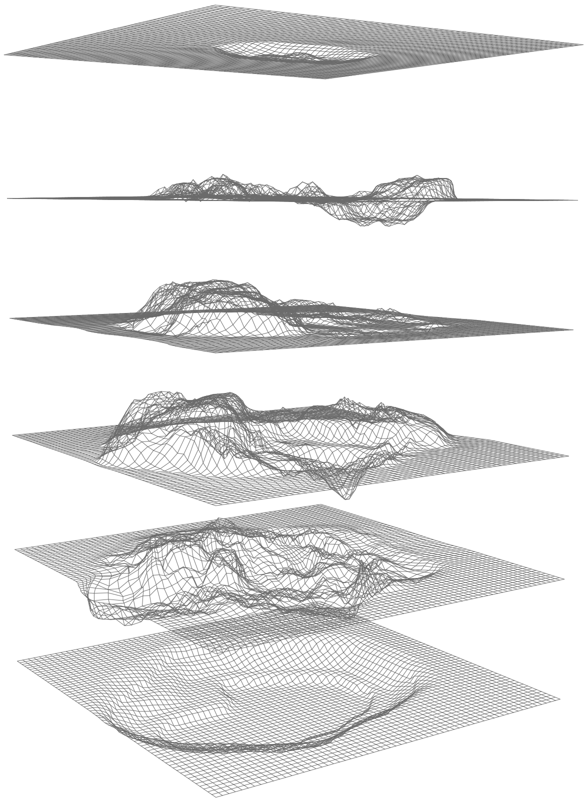


Figure 4: The displacement field can be visualized using a deformed rectangular grid.

Visualization of Neural Networks Using Saliency Maps

Niels J. S. Mørch[+‡] Ulrik Kjems[+] Lars Kai Hansen[+]
Claus Svarer[‡] Ian Law[‡] Benny Lautrup[†] Steve Strother[♮] Kelly Rehm[♮]

[+] Electronics Institute
Technical University of Denmark
DK-2800 Lyngby, Denmark

[‡] Department of Neurology
National University Hospital, Rigshospitalet
DK-2100 Copenhagen Ø, Denmark

[†] Niels Bohr Institute
University of Copenhagen
DK-2100 Copenhagen Ø, Denmark

[♮] PET Imaging Service, Va Medical Center
Radiology and Health Informatics Depts.
University of Minnesota, Minneapolis
Minnesota, 55417, USA

E-Mail : `nmorch@ei.dtu.dk`

ABSTRACT

The saliency map is proposed as a new method for understanding and visualizing the non-linearities embedded in feed-forward neural networks, with emphasis on the ill-posed case, where the dimensionality of the input-field by far exceeds the number of examples. Several levels of approximations are discussed. The saliency maps are applied to medical imaging (PET-scans) for identification of paradigm-relevant regions in the human brain.
Keywords: saliency map, model interpretation, ill-posed learning, PCA, SVD, PET.

1. Introduction

Mathematical modeling is of increasing importance in medical informatics. In bio-medical context the aim of neural network modeling is often twofold. Besides using empirical relations established within a given model, there is typically a wish to interpret the model in order to achieve an *understanding* of the processes underlying and generating the data. This paper presents a new tool for such opening of the neural network "black box".

Our method is aimed at neural network applications where the network is trained to provide a relation between huge, highly correlated, measurements and simple "labels". The measurement could e.g. be a spectrum, an image, or as in our particular case a brain scan volume. The label could be a concentration, a diagnosis etc.

In neural network applications, an important aspect of the training process is the architecture synthesis. An architecturally optimized network supplies structural information about the input field as used by the model, thus giving a qualitative measure of importance.

The output of our new procedure is a "map" *quantifying* the importance (*saliency* c.f. [7]) of each individual component of the measurement (i.e. pin, pixel, or voxel) with respect to the obtained empirical relation. Hopefully, this so-called *saliency map* will assist the modeler in interpreting the model and in communicating the interpretation to the end-user.

In bio-medical context it is often hard (not to say expensive) to gather large samples of data. Hence, if modeling from high dimensional data based on small samples, one faces an extremely ill-posed learning problem and standard practice has been to apply hand crafted tools ("a priori knowledge") for preprocessing and data reduction in order to bring down the dimensionality of the neural network. However, we have recently shown that one may cure this extremely ill-posed problem using straightforward linear algebra *without loss of information* [2], [5]. The scheme achieves *massive weight sharing* [7] by projecting the high dimensional data onto a low dimensional basis spanning the so-called signal space of the training set input vectors. The saliency map is an attempt to visualize this induced geometry and the specific manner in which this geometry is used by the trained network.

As a specific case, we consider modeling of images obtained from Positron-Emission-Tomography

(PET)-scans which is a technique offering 3-dimensional volume measurements of human brain activity. A neural network may be trained using supervised learning on a given training set of PET-scans [2], [5]. We investigate two cases, based on two sets of 64 scans each (8 subjects scanned 8 times): one where the subjects perform an eye movement task according to a graduated (parameterized) paradigm [6], and one where they perform a finger opposition task [12]. In the first case the network is trained to predict the paradigm graduation parameter—the frequency of the saccadic eye movements—using the measured activation patterns in the brain volume as input. In the latter the network is trained to classify the measured activation patterns as rest or activated (i.e. doing the finger opposition task). Since the models are nonlinear, the interpretations are not straightforward. In this particular case the saliency map can be viewed as a tool for visualizing the regions in the brain, which are related most strongly to the specific tasks.

2. The Saliency Map

It is well-known that affine preprocessing [8, 10] can assist training and generalization significantly. Affine preprocessing of an input vector $\boldsymbol{x}_j$ (i.e. an element of the training set of inputs $\mathbf{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_J]$) can be expressed as $\boldsymbol{v}_j = \mathbf{B}^T(\boldsymbol{x}_j - \boldsymbol{c})$. In fact, translating by the training set averaged input vector $\boldsymbol{c} = \bar{\boldsymbol{x}}$ and computing the projection matrix $\mathbf{B}$ from a diagonalization of the input covariance matrix we may obtain $\boldsymbol{v}_j$ as the principal components[1] of $\mathbf{X}$. For simplicity we replace $\boldsymbol{x}_j - \boldsymbol{c}$ with $\boldsymbol{x}_j$ in the following, without loss of generality.

In image or volume processing, where the number of input channels I is often much greater than the number of examples J, a transformation like above can be used to reduce the dimensionality of the data-representation. However, it should be noted that within our scheme for handling extremely ill-posed problems the preprocessing doesn't necessarily reduce the data[2], in contrast to what is often the purpose when employing PCA, but may merely transform the data to a convenient (orthogonal) basis—thus we may have $\operatorname{rank}(\mathbf{X}) = \operatorname{rank}([\boldsymbol{v}_1 \ldots \boldsymbol{v}_J])$. In this way we map the high dimensional input data vector onto a much smaller data vector of *projections*—hence, enforcing relations between elements of the weights connecting input to hidden units of the feed forward neural network, in other words we achieve a massive weight sharing. For a more detailed description see [2], [5]. Spelled out in terms of the neural network this can be written,

$$\begin{aligned} F(\mathcal{W}, \mathbf{B}, \boldsymbol{x}) &= F(\mathcal{W}, \mathbf{B}^T \boldsymbol{x}) \\ &= \sum_a W_a \tanh\left(\boldsymbol{w}_a^T \mathbf{B}^T \boldsymbol{x}\right) \end{aligned} \quad (1)$$

which is now a function of the input $\boldsymbol{x}$ projected on the set of $K \leq \operatorname{rank}(\mathbf{X})$ basis vectors[3] $\boldsymbol{b}_k$ forming the basis $\mathbf{B} = \left[\boldsymbol{b}_1 \ldots \boldsymbol{b}_K\right]$ and a set of weight parameters $\mathcal{W} = \{W_a, \boldsymbol{w}_a\}$. The constrained weights are in turn optimized using a training set[4] $\mathcal{T} = \left\{(\boldsymbol{x}_j, y_j) \,|\, j = 1, \ldots, J\right\}$ by minimizing the cost function with respect to $\mathcal{W}$

$$E(\mathcal{W}, \mathbf{B}, \mathcal{T}) = \frac{1}{J} \sum_{j=1}^{J} \left(y_j - F(\mathcal{W}, \mathbf{B}^T \boldsymbol{x}_j)\right)^2, \quad (2)$$

and we define:

> The saliency of input channel i (or pixel i if $\boldsymbol{x}$ is an image vector) is the change in the cost-function when the i'th input channel is removed.

This removal can be thought of as changing the basis vectors in $\mathbf{B}$, resulting in the new basis $\check{\mathbf{B}}^i$

$$\check{\boldsymbol{b}}^i_{k,i'} = \begin{cases} \boldsymbol{b}_{k,i'} & i' \neq i \\ 0 & i' = i \end{cases} \quad (3)$$

i.e. setting the i'th component[5] of all basis vectors to 0. Introducing this new basis, the model should be retrained to yield a new set of weight parameters $\check{\mathcal{W}}^i$. The saliency of input channel i is therefore

$$\delta E_i = E(\check{\mathcal{W}}^i, \check{\mathbf{B}}^i, \mathcal{T}) - E(\mathcal{W}, \mathbf{B}, \mathcal{T}). \quad (4)$$

If pruning is used to eliminate the effect of noise it should be applied to the full network prior to the calculation of the saliency map, so the retraining after removing the individual inputs conserves the network architecture.

Ideally one could estimate the change in generalization ability [11]. Such an estimate would—given a limited amount of data—be quite inaccurate, and since we only want to use the saliency map for comparing the relative input importance, it seems reasonable to consider only the change in the training error as indicated in equation (4).

Further approximations depend on the specific problem: In image processing the number of input channels (pixels) is often much greater than the number of examples, so that the computational burden of the direct computation of the saliency may be impractical. For such applications we develop

[1] The principal components as obtained from SVD (Singular Value Decomposition), or PCA (Principal Component Analysis). In either case the basis vectors correspond to the eigenvectors of the input data covariance matrix, see [4].

[2] In the sense of loosing information.

[3] See also section 2.1 for a more detailed explanation of the notation.

[4] The outputs are assumed scalar for simplicity.

[5] By the notation $\boldsymbol{b}_{k,i}$ we mean the i'th element of $\boldsymbol{b}_k$.

approximations of the saliency map using an expansion of the cost function. This is further described in section 2.1.

Finally, let us note that the saliency map as such is not confined to the ill-posed learning problem. In more conventional neural network applications, where the number of network inputs I is much smaller than the number of examples J, the saliency is similar to the *sensitivity measure* proposed in [14], [13] and [9], and to the Optimal Cell Damage Scheme suggested in [1]. In this case the removal of a single input may cause a notable change in the optimal weights thus making the I network retrainings essential (in contrast to the ill-posed case, as we shall see).

2.1. The Saliency Map in the Ill-Posed Case

As discussed a significant computational reduction can be obtained by projecting on the set of basis vectors $\mathbf{B}$ spanning the signal space[6] $\mathcal{S}$, if $I \gg J$.

It is easily seen [2], [5] that training in this case preserves signal space, i.e., if the initial weights of a hidden unit are confined to signal space they will stay there during training. This is a consequence of the fact that the cost function is independent of any component of the weight parameters outside signal space, $\mathcal{S}$, regardless of the basis $\mathbf{B}$ used for representing the data, as long as $\mathbf{B}$ spans $\mathcal{S}$.

After preprocessing the neural network is not fed the actual pixel data, but the projection of the images on the basis $\mathbf{B}$. This justifies the notation $F(\mathcal{W}, \mathbf{B}^T \boldsymbol{x}_j)$ for the model, in that the model can be said to be working on the projected data $\boldsymbol{v}_j = \mathbf{B}^T \boldsymbol{x}_j$.

2.1.1. Approximating the Saliency Map

If the number of input channels I is large, the task of retraining I networks—i.e. to compute $\check{\mathcal{W}}^i$ as implied by equation (4)—is immense. In this section some approximations are presented to speed up the computation.

The second order expansion of the cost function with respect to the basis vectors and the weight vector $\boldsymbol{u} = \left[\boldsymbol{w}_1^T \ldots \boldsymbol{w}_A^T \; W_1 \ldots W_A\right]^T$ consisting of all the parameters in $\mathcal{W}$ is given by

$$\begin{aligned} \delta E \;\simeq\;& \sum_{k=1}^{K} \frac{\partial E}{\partial \boldsymbol{b}_k^T}\delta \boldsymbol{b}_k + \frac{\partial E}{\partial \boldsymbol{u}^T}\delta \boldsymbol{u} \\ +\;& \frac{1}{2}\sum_{k=1}^{K} \delta \boldsymbol{b}_k^T \frac{\partial^2 E}{\partial \boldsymbol{b}_k \partial \boldsymbol{b}_k^T}\delta \boldsymbol{b}_k + \frac{1}{2}\delta \boldsymbol{u}^T \frac{\partial^2 E}{\partial \boldsymbol{u} \partial \boldsymbol{u}^T}\delta \boldsymbol{u} \\ +\;& \sum_{k=1}^{K} \delta \boldsymbol{b}_k^T \frac{\partial^2 E}{\partial \boldsymbol{b}_k \partial \boldsymbol{u}^T}\delta \boldsymbol{u}, \end{aligned} \tag{5}$$

where $\delta \boldsymbol{b}_k$ is the change in the k'th basis vector, and $\delta \boldsymbol{u}$ is the change in the optimal weight parameters, due to the changed basis. If the network is fully trained $\frac{\partial E}{\partial \boldsymbol{u}} = \mathbf{0}$ so the second term vanishes[7].

In the ill-posed case, modeling will only be meaningful if the stochastic part of the signal is highly correlated, i.e., the individual pixels are spatially correlated. Thus it can be assumed that the term $\delta \boldsymbol{u}$ roughly scales inversely with the number of inputs, i.e. as $1/I$. We therefore neglect all terms scaling with $\delta \boldsymbol{u}$ yielding

$$\delta E \simeq \sum_{k=1}^{K} \frac{\partial E}{\partial \boldsymbol{b}_k^T}\delta \boldsymbol{b}_k + \frac{1}{2}\sum_{k=1}^{K} \delta \boldsymbol{b}_k^T \frac{\partial^2 E}{\partial \boldsymbol{b}_k \partial \boldsymbol{b}_k^T}\delta \boldsymbol{b}_k, \tag{6}$$

thus eliminating the effect of retraining, effectively estimating $\delta E_i = E(\mathcal{W}, \check{\mathbf{B}}^i, \mathcal{T}) - E(\mathcal{W}, \mathbf{B}, \mathcal{T})$ c.f. equation (4). This is in line with the Optimal Brain Damage scheme [7] for estimating weight saliency and the approximation is indeed supported by the numerical example. Since we compute the saliency for one input channel at a time, the off-diagonal elements of $\frac{\partial^2 E}{\partial \boldsymbol{b}_k \partial \boldsymbol{b}_k^T}$ vanish, so we finally get

$$\delta E_i \simeq \sum_{k=1}^{K} \frac{\partial E}{\partial \boldsymbol{b}_{k,i}}\delta \boldsymbol{b}_{k,i} + \frac{1}{2}\sum_{k=1}^{K} \frac{\partial^2 E}{\partial \boldsymbol{b}_{k,i}^2}\delta \boldsymbol{b}_{k,i}^2. \tag{7}$$

For the two-layer network specified in equation (1), with $h_{aj} = \tanh\left(\boldsymbol{w}_a^T \mathbf{B}^T \boldsymbol{x}_j\right)$ we find[8]

$$\begin{aligned} \frac{\partial E}{\partial \boldsymbol{b}_{k,i}} &= -\frac{2}{J}\sum_{j=1}^{J} \Big[\left(y_j - F(\mathcal{W}, \mathbf{B}^T \boldsymbol{x}_j)\right) \\ &\qquad \cdot \sum_a W_a (1 - h_{aj}^2) \boldsymbol{w}_{a,k} \boldsymbol{x}_{j,i} \Big] \\ &= -\frac{2}{J}\sum_{j=1}^{J} e_j s_{jk} \boldsymbol{x}_{j,i} \end{aligned} \tag{8}$$

where we have introduced the quantities $e_j = y_j - F(\mathbf{B}^T \boldsymbol{x}_j, \mathcal{W})$ and $s_{jk} = \sum_a W_a (1 - h_{aj}^2) \boldsymbol{w}_{a,k}$. By further invoking the Gauss-Newton approximation $\left(\frac{\partial^2 E}{\partial \boldsymbol{b}_k \partial \boldsymbol{b}_k^T} \simeq \sum_{j=1}^{J} \frac{\partial F}{\partial \boldsymbol{b}_k}\frac{\partial F}{\partial \boldsymbol{b}_k^T}\right)$ for least squares problems, see e.g. [7], yielding

$$\frac{\partial^2 E}{\partial \boldsymbol{b}_{k,i}^2} \simeq \frac{2}{J}\sum_{j=1}^{J} s_{jk}^2 \boldsymbol{x}_{j,i}^2, \tag{9}$$

[6] We denote the space spanned by the input vectors $\boldsymbol{x}_j$ in the training set $\mathcal{T}$ by signal space $\mathcal{S} = \mathrm{span}\{\boldsymbol{x}_j\}$.

[7] If we eliminate overfitting by pruning the network, i.e. forcing some parameters $\boldsymbol{u}'$ to $\mathbf{0}$, only the remaining parameters $\boldsymbol{u}^\star = \boldsymbol{u} \setminus \boldsymbol{u}'$ are optimized so that $\frac{\partial E}{\partial \boldsymbol{u}^\star} = \mathbf{0}$. On the other hand, we will generally have $\frac{\partial E}{\partial \boldsymbol{u}'} \neq \mathbf{0}$, which may cause negative estimates of the saliency. This can be explained as follows: If the network models from a subspace of $\mathcal{S}$, called model-space $\mathcal{M}$, one might say that the basis change in (3) perturbs signal space, so that some of the noise eliminated by pruning re-enters $\mathcal{M}$. Sometimes this will allow the model to perform better on the training set, thus yielding negative saliencies. We therefore choose to interpret these as zero.

[8] Again $\boldsymbol{w}_{a,k}$ means the k'th element of $\boldsymbol{w}_a$, and $\boldsymbol{x}_{j,i}$ the i'th element of $\boldsymbol{x}_j$.

and since we remove only one input channel in the basis, i.e. $\delta \boldsymbol{b}_{k,i} = -\boldsymbol{b}_{k,i}$, we get

$$\delta E_i = \frac{2}{J}\sum_{k=1}^{K}\sum_{j=1}^{J} e_j s_{jk} \boldsymbol{x}_{j,i} \boldsymbol{b}_{k,i} + \frac{1}{J}\sum_{k=1}^{K}\sum_{j=1}^{J} s_{jk}^2 \boldsymbol{x}_{j,i}^2 \boldsymbol{b}_{k,i}^2. \tag{10}$$

as the estimate of the saliency map.

3. Ill-posed Example: Modeling from PET images

We now proceed to demonstrate the practical use of the saliency map. Positron-Emission-Tomography (PET) is a way of indirectly measuring the neural activity of different regions of the human brain, resulting in 3-dimensional images. As the dimension of the images is very large, affine preprocessing (projection of the data on the corresponding PCA-basis) is applied, thus reducing the computational requirement of the modeling.

More specifically, we first examined 64 PET-scans of 8 subjects, each scanned 8 times, exposed to 8 different levels of saccadic eye movement activation [6]. We thus analyze $J = 64$ image vectors of $I = 128 \times 128 \times 48 = 768432$ voxels[9].

A two-layer feed-forward neural network was trained to predict the paradigm activation level (the frequency of the saccadic eye movements) from the 64 3-dimensional brain volumes.

An estimated saliency map was computed employing the approximation in equation (10). In figure 1 iso surfaces (surfaces of equal saliency) capturing the most salient voxels are depicted as bright bodies floating in a box. To help localize the salient areas, slices of a corresponding anatomical brain image (an MR scan) are shown on the walls of the box, with the shadows of the salient bodies projected in black. The slices correspond to the middle of the brain, one in each of the three dimensions.

The result is in correspondence with what has been found using other analysis methods—e.g. Statistical Parametric Mapping (SPM), and the Scaled Subprofile Model (SSM)—on the same data [6], [12]. The larger cluster of salient pixels, as seen in the back of the brain, is identified as the *visual cortex.*

To demonstrate the accuracy of the 1st and 2nd order approximations of the saliency, c.f. equation (10), we computed the images shown in figure 2. The first column shows the true change in the cost function[10] for horizontal slices through the volume corresponding the AC-PC[11] level -17mm, AC-PC,

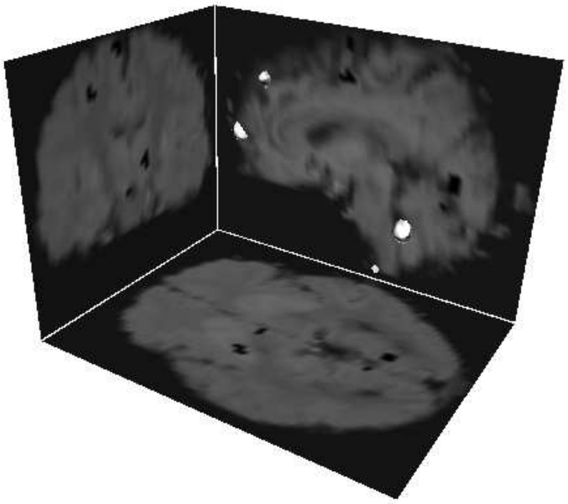


Fig. 1: Using the saliency map to asses paradigm related brain regions in the saccadic eye movement task. The most salient voxels are depicted as iso surfaces (surfaces of equal saliency) here seen as bright bodies floating in a box with slices of a corresponding anatomical brain scan depicted on the walls. Shadows of the iso surfaces are projected in black on the walls. The larger cluster in the back of the brain is the *visual cortex.*

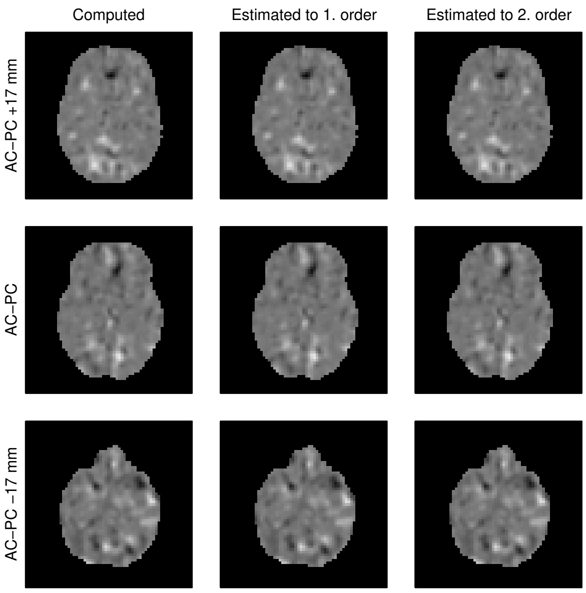


Fig. 2: From left to right: Computed saliency map, 1st order, and 2nd order approximations, all for 3 different slices of the brain. The slices correspond to the AC-PC level -17 mm, the AC-PC level and the AC-PC level + 17mm. Bright areas have high saliencies. In the specific case (I = 34863 pixels) all columns are almost identical—thus validating the approximations. In fact, the 2nd order term seems visually negligible.

and AC-PC + 17mm. This corresponds to expanding E to infinitely high order with respect to $\boldsymbol{b}$. The second and third columns are the 1st and 2nd order

[9] Of these a large portion is masked out, leaving vectors of "only" active 34863 voxels.

[10] Computed as the change in the cost function *without* retraining $\delta E_i = E(\mathcal{W}, \check{\mathbf{B}}^i, \mathcal{T}) - E(\mathcal{W}, \mathbf{B}, \mathcal{T})$, so that only the effects of neglecting the higher order 'pure' $\delta \boldsymbol{b}_k$ terms of (5) and (6) are assessed.

[11] Anterior Comisura - Posterior Comisura, which are easily identified centers in the brain, and thus used for reference.

approximations of (10). It is evident, that even the 1st order term alone is a useful approximation in the case of $I = 34863$ voxels.

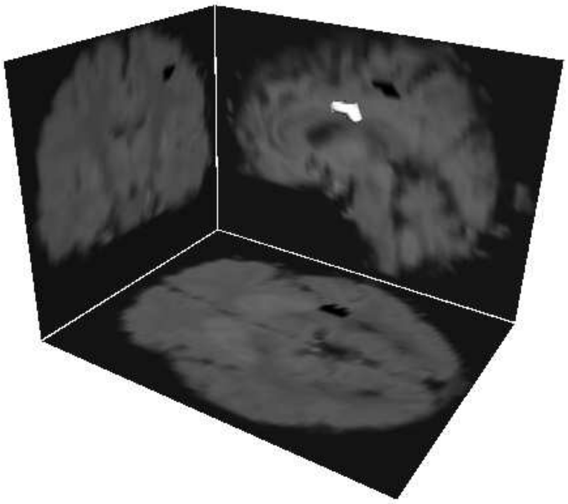


Fig. 3: Saliency map of the finger opposition task. The most salient voxels are depicted as iso surfaces (surfaces of equal saliency) here seen as bright bodies floating in a box with slices of a corresponding anatomical brain scan depicted on the walls. Shadows of the iso surfaces are projected in black on the walls. The salient area identified is the *primary sensory-motor cortex.*

Secondly, the saliency map was computed for a neural network modeling the finger opposition task, which involves areas of the brain controlling motion. The data has previously been analyzed in [12]. Again, 8 subjects were scanned 8 times each, 4 times resting and 4 times doing the finger opposition task. Thus the paradigm is on/off corresponding to a problem of classification[12]. Figure 3 shows the saliency map in a manner similar to figure 1. The method clearly identifies the area known as the *primary sensory-motor cortex.*

Further, we investigated the effect of the dimension of the input-field I, on the approximation (10). For simplicity this is done on a single slice, which is sub-sampled to yield $Q = 9$ datasets with decreasing I. After performing the entire modeling procedure Q times, we measure as a function of I the normalized mean squared error for both the 1st and 2nd order expansions, i.e

$$\begin{aligned} f_1(I) &= \frac{\sum_{i=1}^{I}\left(\delta E_{c,i} - \delta E_{1,i}\right)^2}{\sum_{i=1}^{I}\delta E_{c,i}^2} \\ f_2(I) &= \frac{\sum_{i=1}^{I}\left(\delta E_{c,i} - \delta E_{2,i}\right)^2}{\sum_{i=1}^{I}\delta E_{c,i}^2} \\ \delta E_{1,i} &= \sum_{k=1}^{K}\frac{\partial E}{\partial \boldsymbol{b}_{k,i}}\delta \boldsymbol{b}_{k,i} \\ \delta E_{2,i} &= \delta E_{1,i} + \frac{1}{2}\sum_{k=1}^{K}\frac{\partial^2 E}{\partial \boldsymbol{b}_{k,i}^2}\delta \boldsymbol{b}_{k,i}^2 \\ \delta E_{c,i} &= E(\mathcal{T},\mathcal{W},\check{\mathbf{B}}^i) - E(\mathcal{T},\mathcal{W},\mathbf{B}) \end{aligned} \qquad (11)$$

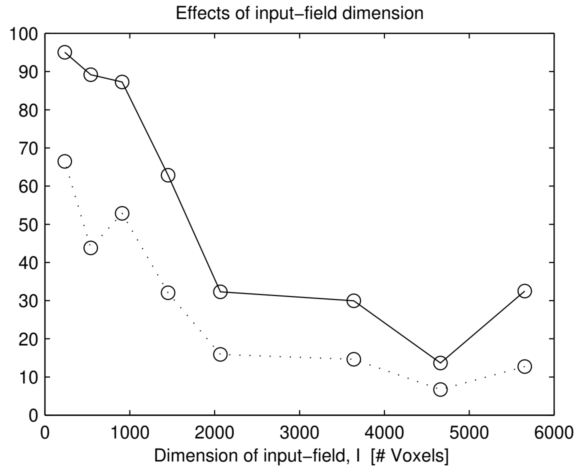


Fig. 4: Normalized mean squared error of the 1st (—) and 2nd ($\cdots$) order approximations of the saliency. With increasing input-field dimension I, the errors decrease—for large I the 1st order approximation suffices.

These quantities are shown in figure 4. We see that the error introduced by the approximations decreases when I gets large. Further, for very large I, the 2nd order term seems negligible. This is in line with the visual impression of figure 2.

Finally, let us note that the saliency map easily computes for linear models as well.

4. Discussion

We have proposed the saliency map as a new method for understanding and visualizing feed-forward neural networks. Furthermore, several levels of approximations have been derived providing significant computational savings. The viability of the approach was demonstrated on a series of 3D brain activation volumes.

Though the emphasis has been on the so-called ill-posed case, the proposed technique can easily be

[12] Note that for classification problems better optimization schemes (costfunctions) exist, see e.g [3].

applied to the more standard setting, i.e. the well-posed case.

5. Acknowledgments

This research has been supported by the Danish Research Councils for the Natural and Technical Sciences through the Danish Computational Neural Network Center CONNECT, and the US National Institutes of Health's Human Brain Project through grant DA09246.

References

[1] T. Cibas *et al.*, "Variable selection with optimal cell damage," *Proceedings of the International Conference on Artificial Neural Networks, pp. 727–730*, 1994.

[2] L. K. Hansen, B. Lautrup, I. Law, N. Mørch, and J. Thomsen, "Extremely ill-posed learning," CONNECT *Preprint.* Available via anonymous ftp `ei.dtu.dk : dist/hansen.ill-posed.ps.Z.`, Aug. 1994.

[3] M. Hintz-Madsen *et al.*, "Design and evaluation of neural classifiers - application to skin lesion classification," *To appear: 1995 IEEE Workshop on Neural Networks for Signal Processing (NNSP'95).* Available via anonymous ftp `ei.dtu.dk : dist/1995/hintz.nnsp95.ps.Z`, 1995.

[4] J. E. Jackson, *A User's Guide to Principal Components.* Wiley Series on Probability and Statistics, John Wiley and Sons, 1991.

[5] B. Lautrup, L. K. Hansen, I. Law, N. Mørch, C. Svarer, and S. Strother, "Massive weight sharing: A cure for extremely ill-posed problems," in *Proceedings of Workshop on Supercomputing in Brain Research: From Tomography to Neural Networks, HLRZ, KFA Jülich, Germany*, (H. J. Hermann, D. E. Wolf, and E. P. Pöppel, eds.), pp. 137–148, Nov. 1994.

[6] I. Law *et al.*, "A characterization of the frequency related cerebral response during sensory-guided saccades," *In preparation*, 1995.

[7] Y. Le Cun, J. S. Denker, and S. Solla, "Optimal brain damage," *Advances in Neural Information Processing Systems 2, pp. 598–605*, 1990.

[8] Le Cun, Y., I. Kanter, and S. Solla, "Eigenvalues of covariance matrices: Application to neural-network learning," *Physical Review Letters*, vol. 66, Number 18:pp 2396–2399, May 1991.

[9] J. Moody, "Prediction risk and architecture selection for neural networks," in *From Statistics to Neural Networks, Theory and Pattern Recognition Applications*, (V. Cherkassky, J. H. F. H., and H. Wechsler, eds.), pp. 147–165, Springer Verlag, 1992.

[10] J. S. Orfanidis, "Gram-schmidt neural nets," *Neural Computation*, vol. 2, pp 116–126, 1990.

[11] M. W. Pedersen, L. K. Hansen, and J. Larsen, "Pruning with generalization based saliencies: γOBD, γOBS," *Submitted to: Advances in Neural Information Processing Systems (NIPS95).* Available via anonymous ftp `ei.dtu.dk : dist/1995/with.nips95.ps.Z`, 1995.

[12] S. C. Strother, J. R. Anderson, K. A. Schaper, J. J. Sidtis, J. S. Liow, R. P. Woods, and D. A. Rottenberg, "Principal component analysis and the scaled subprofile model compared to intersubject averaging and statistical parametric mapping: I. "Functional connectivity" of the human motor system studied with [15-o]water pet," *Journal of Cerebral Blood Flow and Metabolism*, vol. 15, pp. 738–753, 1995.

[13] J. Utans and J. Moody, "Principled architecture selection for neural networks: Application to corporate bond rating prediction," *Advances in Neural Information Processing Systems 4*, 1991.

[14] J. Utans and J. Moody, "Selecting neural network architectures via the prediction risk: Application to corporate bond rating prediction," *Proc. First International Conference in Artificial Intelligence Applications on Wall Street*, 1991.

Appendix B

Program Documentation

B.1 The volutil Package

The package volutils contains a collection of UNIX programs written in ANSI C, for registration and resampling of three dimensional volume data. The non-linear warp program is the core of the package; the other utilities are useful for practical application of the non-linear warp program to a functional data set. The utilities are all command-line based, which in the authors point of view, is the best suited for interface to existing analysis systems. Included is also an interactive volume browser for Matlab 5. Users are encouraged to utilize the volio C package which has been designed for easy implementation and development of applications in C for handling volumes in multiple data and file formats. All programs are available through anonymous ftp from eivind.imm.dtu.dk/pub/volio and are documented on-line at http://eivind.imm.dtu.dk/staff/kjems/volio. The programs have been tested on the Linux, Irix, HP-UX, Sun-OS and Alpha-OS operating systems. Documentation of the Unix utilities comprises man-pages which are reproduced in the following pages.

Program	Description	Page
compute_warp	Compute a 3D non-linear deformation ("warp") of a volume to match a template.	96
warp_reslice	Applies deformation field and linear transformation	104
convert_vol	Change datatype, scale intensity, compute min/max	113
covar	Compute covariance matrix for a set of volumes	114
file2vapet, file2ana, file2xprime	Convert a file to a different file format	110
flip	flip a volume along X-, Y- or Z-axis	116
convert_vol	Change datatype, scale intensity, compute min/max	113
gauss_vol	Apply Gaussian smoothing kernel to a volume	117
headerinfo	Show volume header information	111
histo_vols	Compute 2D intensity histogram of 2 volumes	108
resize_vol	Resize a volume (add/remove space)	112
subsample	Subsample a volume by an integral factor using sinc-interpolation	109
warp_reconcile	Reconcile A→B and B→A warp fields	118

NAME

compute_warp - Compute a 3D non-linear deformation ("warp") of a volume to match a template.

SYNOPSIS

compute_warp [-h] | [*options-file {keyword=value}* **]**

DESCRIPTION

compute_warp computes a non-linear deformation for inter-subject normalization of brain scans. The deformation field consists of a set of displacement vectors sampled on a coarser grid. **optionsfile** contains the parameters for the warp. If '-' is specified as optionsfile, no optionsfile is used.

The produced deformation grid is fed to the program **warp_reslice** which performs the actual reslicing of the volumes.

OPTIONS

-h Display help and a list of recognized keywords with a short description. This output may seem a little cryptical for most people, hence this man-file.

keyword=value
: Most major options are controlled using keyword=value assignments in the optionsfile. Options may also be specified using keyword=value pairs on the command line after the optionsfile. Options specified on the command line overrule any definitions in the optionsfile. Note that the keyword=value pair on the command line must be enclosed in quotes if there are spaces in the string, e.g.

```
compute_warp warp.opt level=info "alpha=0.1 0.05"
```

to read options from the file warp.opt and to set the verbosity to "info" and to set the alpha parameter. Allowed keyword/value pairs are described the the later section FORMAT OF THE OPTIONSFILE.

INTRODUCTION TO MRI ALIGNMENT

This program is only one of a series of programs designed for performing non-linear registration (aka: warp) of a set of Magnetic Resonance (MRI) volumes of the human brain. The program is designed for achieving accurate inter-subject alignment of functional brain images by deforming corresponding co-registered purely anatomical (MR) scans.

The program takes two 8 bit MRI volumes and produces a set of displacement vectors (comprising a displacement field). The displacement field provides 3-dimensional vectors at locations in the template volume. The vector at a given location in the template volume is interpreted as the displacement of the anatomically corresponding location in the subject volume (aka: data volume). The displacement vectors are not computed at every voxel location in the template but on a coarser grid centered on the template volume's field of view.

A volume's field of view is the entire collection of space occupied by the voxels in the volume.

Typically the displacement vectors are computed on a 4 times coarser grid so that there exists one displacement vector for each 4x4x4 cube of template voxels. This means that if the template volume is 256x256x160 voxels, the displacement field is going to consist of 64x64x40 vectors centered over 4x4x4 segment blocks in the template volume.

When the displacement field is applied (e.g. using **warp_reslice** or **apply_warp**) displacement vectors are needed for locations in between the locations of the displacement vectors. Tri-linear using the 8 surrounding vectors in the grid of displacement vectors resolves this.

PROCEDURE FOR NON-LINEAR REGISTRATION

This program has been designed and tested for t1 weighted MR images of the human brain. It has been tested on 20ms double spin echo images as well, but t1 is preferable since this modality has the best gray/white matter contrast. The resolution should be better than 1.1mm in at least two dimensions.

The entire procedure for performing non-linear registration of a functional data-set using corresponding anatomical (MR) images consists of the following data steps:

a. Raw MRI's should be corrected for scanner inhomogeneity.

b. Strip skull & scalp

c. Affine alignment of stripped MRI (b) to Talairach template

d. Non-linear alignment of "affined" subject (c) to template

e. Rigid body (6 parameter) alignment to subjects' stripped MRI (intra-subject alignment) for each functional scan

f. Reslicing of functional data to template space combining rigid body (e), affine (c) and non-linear (d) transformations.

Ad. a.: Some 3D MR acquisition protocols produces severe inhomogeneity effects, which are clearly visible in the scans as a varying "illumination" of different regions of the scan. This should be corrected for prior to any further processing.

Ad. b.: Brain stripping is quite often done manually. Some reports have been made about accurate semi- or full-automatic procedures. If the MR scanner acquires more than one modality a multi-channel threshold combined with appropriate morphological operations may do the job.

Ad. c.: The affine alignment is computed f.x. by the use of Roger Wood's AIR package (the **align_linear** program). The **warp_reslice** program accepts the AIR format registration files (as well as free style text format). An affine alignment is one that preserves straight lines straight, (but parallel lines do not necessarily remain parallel). An affine transformation is conveniently expressed as a 4x4 matrix. If image coordinates are represented in homogeneous coordinates (4 dimensional representation) the affine transformation from one volume's coordinate system to the other is represented by the matrix multiplication of the 4x4 transformation matrix with the 4x1 coordinate vector. In order for the transformation to be affine some restrictions apply on the transformation matrix so that the matrix is determined from 12 parameters. Therefore the affine transformation is also labelled the "12 parameter transformation". The 12 parameters take care of rotation, translation and up/down scaling in size in any directions.

Ad. d.: This is the step handled by **compute_warp** ! The non-linear transformation takes care of what remains to be fixed after the 12-parameter affine alignment has been performed. This means that the subject's MR volumes need to be resampled according to the initial 12 parameter alignment prior to computing the non-linear warp.

Ad. e.: The rigid-body intra-subject alignment is handled by e.g. Roger Wood's AIR package. The AIR package may be obtained from

http://bishopw.loni.ucla.edu/AIR3/index.html.

The intra-subject may, similar to the affine transformation, be expressed as a matrix multiplication in homogeneous coordinates only now the matrix is determined from only 6 parameters, 3 for specifying a translation, and 3 for arbitrary rotation.

Ad. f.: The functional scans should always be resampled as few times as necessary since image resolution otherwise may be degraded. The program **warp_reslice** is designed to perform the entire chain of image transformations in one step combining the 6 parameter functional-to-structural, the 12 parameter and the non-linear transformations in one reslicing operation.

HOW DOES THE PROGRAM WORK?

The program uses a hierarchical algorithm for computing the displacement field. The template and data volumes are subsampled to lower resolution and the displacement field is computed on the coarse resolution first. The resolution is increased when the displacement field has converged. The displacement field is made up of fewer vectors in the lower resolution - typically one maintains 1 displacement vector for each 4x4x4 voxels.

For example, for warping a 256x256x160 sized data volume into a template (which has to be same size!) one would typically use three resolution steps: 1/4, 1/2 and full (1/1). In the coarsest resolution step the data and template volumes are subsampled to 64x64x40 voxels and we choose one vector for each 4x4x4 block of voxels (yielding 16x16x10 displacement field vectors).

After 4-8 iterations the displacement field has converged and the program switches to the second resolution step of 128x128x80 voxels. Once again we choose one vector for each 4x4x4 block of voxels now yielding 32x32x20 displacement field vectors. The previous 16x16x10 displacement field is now super-sampled to 32x32x20 using Tri-linear interpolation. And now the program iterates 4-8 times in the second resolution step. Finally in the 3rd resolution step the original 256x256x160 data and template volumes are used and the displacement field is again super-sampled, now to 64x64x40 vectors. The program iterates 4-8 times in the highest resolution and saves the resulting 64x64x40 displacement field.

So what happens in the iterations within a resolution step? Each displacement vector's block of voxels (4x4x4 in the above example) is shifted locally an integer amount of voxels and the optimal shift is stored as the new displacement vector. The optimal shift is identified by evaluating a cost function at each shift location

C = voxel similarity function + alpha * neighbor-connection

The first term is low when there is good correspondence between voxel values in the template box and the subject box of voxels according to a voxel similarity function. This function may be chosen to be either Least Squares, Modified Least Squares, Correlation, or Mutual Information.

The second term determines how much the displacement vector should be affected by neighboring displacement vectors. This term is low when the displacement vector is similar to the average of the 6 nearest neighboring vectors. This term acts more and more as a smoothness enforcer with increasing values of the parameter alpha.

After each iteration the displacement field is smoothed with a low-pass filter in order to further enforce smoothness. This smoothing is performed independently in the X, Y and Z components of the displacement field vectors using a Gaussian kernel.

THE RECONCILED WARP

As a final regularizer for the warp, the displacement field is reconciled with the inverse mapping. Now, what does this mean ? The program actually computes two warps: One from data to template and one with the role of data and template volumes reversed. Ideally after each iteration, the transformation found from data to template should be the inverse of the transformation from template to data. In practice this is not quite the case. In order to reduce error, these two ideally inverse mappings are corrected using a "reconciling operation" which means that the two displacement fields are adjusted so that they look more like each other's inverses.

WHAT SIMILARITY MEASURE TO USE

Mutual information is usually a hit. May produce "strange" warps if not regularised enough. It is always a good idea to inspect the warped volumes and compare against the template volumes to ensure that the warped brain still "looks like a normal brain".

Least squares (0) is faster and works well if you are sure to have the same scaling of voxel intensities in all scans. If there are intensity problems present, you may see problems like white matter being mapped to gray (squeezing the gray matter) or similar problems.

Correlation works well if each scan has it's own global intensity scaling factor.

Local least squares works well if you have problems with local inhomogeneities. It puts more emphasis on matching the edges.

HOW TO DETERMINE PARAMETERS

Most of the parameters don’t really need to be set. You may want to chance the resolution steps

INSPECTING THE WARP RESULT

There are several ways of evaluating the warp result. One is to compute the warped data volume and compare against the template. The shape of the warped brain should resemble the template and some of the fissures should be matched, some are not. An interactive 3-panel volume browser is recommended for this (e.g. the Matlab program shv.m obtainable from ftp://eivind.imm.dtu.dk/pub/volio/voliomatlab.tar.gz).

Secondly the displacement field may be inspected,- this should look smooth and without fast transitions. The Jacobian volume can be computed from the displacement field. (The Jacobian is the determinant of the derivative matrix of the vector field. If the displacement field is the vector function

```
 F(x,y,z)=(vx(x,y,z), vy(x,y,z), vz(x,y,z))
```

the Jacobian is

```
              | dvx/dx, dvy/dx, dvz/dx |
J(x,y,z) =    | dvx/dy, dvy/dy, dvz/dy |
              | dvx/dz, dvy/dz, dvz/dz |
```

Jacobian values that are either high (> 4 or so indicating 4 times expansion of the volume) or low (< 0.25 indicating 4 times compression) are "bad" in the sense that they indicate an unrealistic deformation. The jacobian may be computed using the matlab function below. The program uses the loadvol MEX function obtained from the voliomatlab package (ftp://eivind.imm.dtu.dk/pub/volio/voliomatlab.tar.gz).

```
function [jac,hjac]=getjac(file)
% [jac,hjac]=getjac(file)
%
% Computes the jacobian from the displacement field in "file"
%
% The cmpix is automatically obtained from the header
% The returned header can be used to save the jacobian volume
htrf=loadhdr(file);
trf=double(loadvol(file));
sz=htrf.size;
x1=1:sz(1)-1; x2=2:sz(1); y1=1:sz(2)-1; y2=2:sz(2);
z1=1:sz(3)-1; z2=2:sz(3);
dtrfx=(trf(x2,:,:,:)-trf(x1,:,:,:))/htrf.cmpix(1);
dtrfy=(trf(:,y2,:,:)-trf(:,y1,:,:))/htrf.cmpix(2);
dtrfz=(trf(:,:,z2,:)-trf(:,:,z1,:))/htrf.cmpix(3);
```

```
dd=zeros([htrf.size(1:3)-1,3, 3]);
dd(:,:,:,:,1)=(dtrfx(x1,y1,z1,:)+dtrfx(x1,y2,z1,:)...
   +dtrfx(x1,y1,z2,:)+dtrfx(x1,y2,z2,:))/4;
dd(:,:,:,:,2)=(dtrfy(x1,y1,z1,:)+dtrfy(x2,y1,z1,:)...
   +dtrfy(x1,y1,z2,:)+dtrfy(x2,y1,z2,:))/4;
dd(:,:,:,:,3)=(dtrfz(x1,y1,z1,:)+dtrfz(x2,y1,z1,:)...
   +dtrfz(x1,y2,z1,:)+dtrfz(x2,y2,z1,:))/4;
dd=permute(dd, [4, 5, 1, 2, 3]);
ii = eye(3);
jac=zeros(htrf.size(1:3)-1);
for x=x1,
  for y=y1,
    for z=z1,
      jac(x,y,z)=det(dd(:,:,x,y,z)+ii);
    end,
  end,
  disp(sprintf('%4.1f', x/max(x1)*100))
end
hjac = htrf;
hjac.rank = 3;
hjac.size=size(jac);
hjac.cmpix=htrf.cmpix(1:3);
hjac.min=num2str(min(min(min(jac))));
hjac.max=num2str(max(max(max(jac))));
hjac.filename1='';
```

FORMAT OF THE OPTIONSFILE

The optionsfile is an ASCII file, each line assigns a value or a list of values to a keyword. Anything after '#' on a line is ignored.

Many parameters take a list of values. This is because you may want to use different parameter settings for the different resolution steps. The first resolution step uses the first parameter value specified to all keywords. The program iterates the displacement field in a resolution step until a convergence criteria is met (either maximum number of iterations set via **max_repeat** or the relative change of the cost function is too small, set via **min_rel_change**). Once convergence is reached, the second parameter value of all the keywords is used. You may think of the parameter values in the options file as rows of a matrix, and the program uses a column of the matrix for each resolution step. Note that if a parameter has too few values, the LAST value will be re-used. This means that the keyword with the largest number of values will determine the total number of resolution steps to be applied to the warp.

basedir = string

Base directory to prepend all file names.

templatefile = string list

Name of the template volume. The template volume is the target of the warp. Note that if more than one file is specified the program searches through the list for volumes to use for subsampled versions of the template volume (the original high-res template volume has to be the first in the list).

datafile = string list

Name of the data (aka subject) volume. The data volume is the volume to be warped. Note that if more than one file is specified the program searches through the list for volumes to use for subsampled versions of the data volume (the original high-res template volume has to be the first in the list).

fileformat= string

Determines the default output file format (see volio.h), 0 (default) same as template, 2=VAPET format, 3=ANALYZE format, 4=XPRIME format. The program automatically determines input formats.

outfile_field = string

The name of the generated displacement field data-to-template.

outfile_inv_field = string

The name of the generated reconciled inverse displacement field data-to-template. This is not computed if the reconciled warp feature is disabled (using "no_reconcile=1").

outfile_warped=string

If specified the program will at the end compute the warped data volume and write it to this file. Note that this file is resampled using nearest neighbor sampling which is sub-optimal in most uses. Please use **warp_reslice** which is capable of Tri-linear and sinc-interpolation.

outfile_inv_warped=string

If specified the program will at the end compute the warped template volume and write it to this file. Note that this file is resampled using nearest neighbor sampling which is sub-optimal in most uses. Please use **warp_reslice** which is capable of Tri-linear and sinc-interpolation. This is not computed if the reconciled warp feature is disabled.

logfile=string

Technical information about the warp's inner workings goes into this file.

dumpfile=string

Computational data about the convergence of the warp goes into this text file. The file consists of one row per iteration. The data is

cost_sumsq, cost_sumsqsc, cost_entropy, cost_mutinfo, cost_corr

and 5 more for the inverse mapping if computed.

level=string
Controls amount of output to stdout. Choose from 'silent', 'std', 'info', and 'debug' for least to most output.

method=integer list
Selects the voxel similarity measure to apply. 0: least square, 1: squared scaled, 2: local mean subtracted least squares, 3: max mutual information, 4: min.entropy (not recommended - mathematically unstable), 5: correlation coefficient.

no_reconcile=integer list
Set this to non-zero to disable the reconciled warp feature. Default is enabled.

autoreg=integer list
Set this to non-zero to enable the automatic on-line setting of the regularizer alpha. Default is disabled.

alpha = float list
Selects the amount of neighborhood connection strength. High values gives more restrained warps. Use values of approximately 0.1 for the Mutual Information, 10-50 for the least squares and correlation coefficient similarity functions.

d_filter_fwhm = float list

Controls amount of brute-force smoothing of vector field in between iterations. The parameter sets the FWHM (measured in vector grid distances) of the Gaussian kernel used for the smoothing.

search = integer list
Set to 1,2,or 3. Controls the size of the local search-region for updating vectors in each iteration. Searches plus minus this many voxels, 1=search 3*3*3=27 places, 2=search 5*5*5=125 places, 3=search 7*7*7=343 locations. 2 is recommended and default.

segx=integer list

segy=integer list

segz=integer list
Sets the number of voxels per displacement vector. Typically 4x4x4.

resx=integer list

resy=integer list

resz=integer list
Sets the resolution to use.

NAME

warp_reslice - Applies deformation field and linear transformation

SYNOPSIS

warp_reslice [**-i** *infolevel*] [**-n** | **-l** | **-p** *sx sy sz*] [**-d** *dx dy dz*] [**-c** *cx cy cz*] [**-a** *E2_airfile*] [**-b** *E2_ascii_transfile*] [**-e** *E2_matrix_string*] [**-g** *E1_matrix_string*] [**-o** *E1_ascii_transfile*] [**-j** *inputvol*] [**-f** *displacement_vector_field*] [**-t** *templateexample*] [**-k** *sx sy sz*] [**-m** *cmx cmy cmz*] **outputvol**

DESCRIPTION

Applies a 3D deformation field to a volume and combines the non-linear warp with a linear transformation matrix. This is a general purpose volume data reslice program, which supports VAPET, ANALYZE and XPRIME volume file formats.

OPTIONS

-i level
: silent, std, info or debug for control amount of output

-n Use nearest neighbor interpolation.

-l Use trilinear interpolation.

-p sx sy sz
: Use sinc interpolation using a kernel size of sx * sy * sz. A Hann weigthing filter is applied.

-a airfile
: Reads in a binary .air file as generated by alignlinear. The program checks for differences in endian architechture. The matrix is for cubic template voxels as output by the AIR alignlinear program. This is the E2 matrix.

-b asciitransf2
: Read in a 3x4 or 4x4 transformation matrix from an ascii file. This is the E2 matrix.

-o asciitransf1
: Read in a 3x4 or 4x4 transformation matrix from an ascii file. This is the E1 matrix.

-g matrixstring
: Get the 3x4 or 4x4 E1 transformation matrix directly from this string. The matrix E1 is transversed row by row. The matrix defines the relationship between homogeneous volume index coordinates in the input volume and in the 12 parameter normalized volume:

E1 * output volume coordinates = template index coordinates

E2 * template index coordinates = reslice index coordinates

Note that this is different from the .air convention where the template coordinates refer to cubic voxel sizes.

-e matrixstring
: Get the 3x4 or 4x4 E2 transformation matrix directly from this string. Like -g above.

-j inputvol
: Use this file as the source file for reslicing. If no inputvol is given, the airfile is searched for the name of the reslice file.

-f dispfield
: Volume containing the displacment vectors. If not specified, zero vectors will be applied.

-t templateex
: This is the name of a file which defines voxel spacing of the template volume. Use this volume to get the pixel size for reslicing (instead of using standard file from air file). This will override any naming of the template file given in the .air file (-a option)

-m cmx cmy cmz
: Specify the template voxel spacing here (overriding any other defs).

-k sx sy sz
: In stead of naming a template file, you can specify the template dimensions here (overriding other defs). **OPTIONS CONTROLLING RESLICING OF OUTPUT FILE:**

-d dx dy dz
: Downsample the image by these factors in the x, y and z directions. This is when you want to resample the file over the same field of view as is covered by the template and the displacement map.

-z sx sy sz
: Use these volume dimensions for the output volume. Voxel spacing adjusted accordingly so that the same field of view as the template is covered, unless -c specified as well)

-c cx cy cz
: Use this voxel spacing for the output volume. (dimension adjusted accordingly unless -z specified also)

-h Display this help

All volio.h file and data formats are supported.

ABOUT WARP_RESLICE

Applies a 3D deformation field to a volume and combines the non-linear warp with a linear transformation matrix (as produced by the e.g. the AIR registration package, see http://bishopw.loni.ucla.edu

This is a general purpose volume data reslice program. For non-linear registration of PET using MR scans of each individual subject, the following setup is assumed:

```
  6 par           12 par          nonlinear         Post
  linear          linear          warp              trans/scale
PET --> Subject MRI -->  Template MRI --> Template MRI  -->  Aligned PET
  \_______________________/      \___________/      \___________/
             |                         |                  |
        12 parameter linear      Displacement        Translation
         transformation              field
```

The PET image(s) is registered with the corresponding MR scan using a 6 parameter rigid body reorientation. The subject MR scan is then aligned to the Talairach (or whatever) template MR scan, using a two step procedure: First the subject MR is aligned to the template using a 12 parameter linear affine transformation (translation, rotation and scale in any 3 directions). The nonlinear warp transforms the 12 parameter resliced subject MR to match the template.

This program is able to perform the entire reslice operation, given a transformation matrix and the volume of displacement vectors.

The deformation field is represented as a 4 dim. volume, with the last (outer) index being 0, 1 or 2 for X, Y or Z coordinate of the displacement vector. The displacements are stored in units of cm's. The template volume covers the same field of view as the template volume but may be in lower resolution by any integral factor in each 3 dimensions (i.e. as produced by compute_warp).

The program will resample the input volume 'inputvol' over a field of view that may be different from the field of view covered by the displacement field.

Define the following coordinates (measured in image indices, first voxel is (0,0,0))

```
Subject PET:                             p = (px, py, pz, 1)
12 parameter normalized MR:              m = (mx, my, mz, 1)
Template:                                t = (tx, ty, tz, 1)
Aligned PET:                             u = (ux, uy, uz, 1)
Displacement field vector                F(t) = (F1(t), F2(t), F3(t), 0)
12 par. AIR transformation 4x4 matrix    E2
Output translation 4x4 matrix            E1
```

The program generates the following coordinate shift:

```
  p = E2*m
  m = t/t(4) + F(t)
  t = E1*u
```

Note that if you do not wish to apply a deformation field but only the matrix transformation E1 or E2 you do not need to specify any information regarding the template volume. You will need to specify the total dimensions and voxel size of the output volume.

If you do wish to apply a deformation field, you will need to specify the template size and voxel size, since the displacement field covers the same field of view as the template file.

BUGS

Please tell me if you find any!

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>, see also http://eivind.imm.dtu.dk/staff/kjems

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

NAME

histo_vols - Compute 2D intensity histogram of 2 volumes

SYNOPSIS

histo_vols [**-i** *signal_level*] [**-d**] volume1 volume2 [histogramfile]

DESCRIPTION

Converts volume1 and volume2 to byteformat, and generates a 2D grey-level co-occurrence histogram from the grayscale intensities in the two volumes. The histogram is 256x256, x-axis is volume1 intensity, y-axis is volume2 intensity. If 'histogramfile' is specified the histogram is saved as a 2D volume file, data type LONG.

At the same time, several voxel similarity measures are computed. The program computes:

```
 mean square          (a-b)^2
 scaled mean square   (a-b*k)^2  k=sqrt(Var(a)/Var(b))
 entropy              sum p(a,b) log p(a,b)
 mutual information   sum p(a,b) log (p(a,b)/p(a)p(b))
```

where p(a,b) is the histogram normalized so that the sum over a and b is 1.

OPTIONS

If -d is specified, the program writes out the computed similarity measures in a single row of text so that sequential calls may be concatenated into a matrix.

LIMITATIONS

The input volumes must be 8 bit unsigned integer. VAPET, ANALYZE, and XPRIME file formats are supported.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

SUBSAMPLE() SUBSAMPLE()

NAME

subsample − Subsample a volume by an integral factor

SYNOPSIS

subsample [**−f** *fx fy fz*] [**−s** *sx sy sz*] [**−i** *signal_level*] inputfile outputfile

DESCRIPTION

Subsamples a volume by an integral factor. Subsampling means picking every n'th voxel out out the original volume. Sinc interpolation with a Hann weighted kernel is performed to avoid aliasing. Options are:

−f factorx..z
Specify subsampling factor

−s sinc_size
Specify sinc kernel size in units of the output voxel size. Should be an odd number. Default is 7.

−i level
Spec. report level (DEBUG, STD, INFO, QUIET)

LIMITATIONS

All data formats are supported, although the volume is converted to 32−bit floats during the subsampling process, so precision is lost for 64−bit doubles. VAPET, ANALYZE, and XPRIME file formats are supported.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

NAME

file2vapet, file2ana, file2xprime – Convert a file to a different file format

SYNOPSIS

```
file2vapet [−u −y −n] files
file2ana [−u −y −n] files
file2xprime [−u −y −n] files
```

DESCRIPTION

Converts files to VAPET/ANALYZE/XPRIME file format.

OPTIONS

−u to mirror images up/down

−y yes to all questions

−n no to all q's

−a do NOT append .vol to filename after stripping extension (**file2vapet** and **file2xprime** only)

−s do NOT append strip extension prior to adding .hdr and .img to filenames after stripping extension (**file2ana** only)

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

HEADERINFO() HEADERINFO()

NAME

headerinfo − show volume header information

SYNOPSIS

headerinfo [**−i** *reportlevel*] [−2] [−s] [−f] [−p] [−q] files

DESCRIPTION

Displays header info of the listed files. A common use is

```
headerinfo *.hdr | grep name
```

to list name entries in all .hdr files.

OPTIONS

−2 : ensure power of 2 dim. −p : print X Y Z cmpix_x/min(cmpix) cmpix_y/min(cmpix) cmpix_z/min(cmpix) string (for volren) −q : print X Y Z cmpix_x cmpix_y cmpix_z string (for volren) −f : print filename before each line (for use with grep) −s : make short description only −i : reportlevel is silent,std,info,debug

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

NAME

resize_vol − resize a volume (add/remove space)

SYNOPSIS

resize_vol [**−d** *newx..z*][**−a** *background*][**−s** *oldstartx..z* | **−b** *newstartx..z*] inputvol outputvol

DESCRIPTION

The volume is resized by stripping/adding boundary. Background may bespecified where the output image exceeds the input image (default is 0). Use −s and −b to specify where in the output image the input image should start. If neither −s or −b is specified the new image is centered around the old center. Use −b when reducing image size, and −s when expanding (adding air around the image)

−d newx newy newz
The size of the output volume.

−a background
The background fill−in value when increasing volume size.

−b x y z
starting index (x,y,z) is relative to the output volume (i.e. where does 0,0,0 in output image come from in input image).

−s x y z
starting index (x,y,z) of is relative to the input volume (i.e. where does 0,0,0 in input image go in output image).

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

CONVERT_VOL() CONVERT_VOL()

NAME

convert_vol − change datatype, scale intensity, compute min/max

SYNOPSIS

convert_vol <options> infile [outfile]

DESCRIPTION

Utility for changing the data format in a volume. Scaling and offset is also possible.

OPTIONS

−i level
Set report level std, info, silent

−q # Autoscale to match 0..# in destination

−a # Subtract this from input image

−s # Multiply this on image AFTER above subtraction

−p # Multiply this on image BEFORE above subtraction

−c compute and display max and min intensity and exit

−o *format*
Specify output format to convert to *format* = byte, ushort, sshort, float, double Default setting is to scale image 1:1 to byte format.

LIMITATIONS

Input file format can be VAPET, XPRIME, ANALYZE. All volio input data types are accepted, i.e. byte, ushort, sshort, float, double.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

NAME

covar − Compute covariance matrix for a set of volumes

SYNOPSIS

covar [−i level] [−l scalelist] [−m maskfile] [−s meanvol] [−o outputmatrix] [−d] [−p type] file1 file2 ... fileN

DESCRIPTION

Computes N x N empirical covariance matrix between the specified volumes. The matrix is not normalized, i.e. if the files 1..N make up the columns of a data matrix X then the output is (X−<x>)'*(X−<x>) where <x> is the mean across columns of X.

The program is designed to handle large volumes so volumes are read in multiple times. There has to be memory to store 3 volumes, one in internal rep. (for the mean volume of the <x>'s) and two in input raw format/ I.e. if the volumes are byte format, and the mean is stored as 16 bit, the total memory requirement is 2 byte volumes plus one 16 bit volume, i.e. four times one input volume. The internal representation for the mean is chosen automatically, but you can use the −p option to manually specify the internal representation for the mean. Note that <x> is stored as the sum of file1..N i.e. it is integer when the input format is too (and no scaling is applied).

OPTIONS

−d compute only diagonal entries of covariance matrix.

−x max
: use no more than 'max' megabytes of memory.

−p type
: specifies the datatype that should be used for internal computation T_BYTE=1, T_USHORT=2, T_SSHORT=3 , T_INT32=4, T_FLOAT=5, T_DOUBLE=6

−m maskfile
: The input 'maskfile' tells covar which voxels to use. This saves memory because the mean vectors are only stored for non-zero mask voxels.

−s meanvol
: specifies where to save the computed mean image. Note that for integer datatypes, the output is the sum rather than the mean.

−l scalelist
: is an ASCII string containing 1 float per volume (sep by spaces). Each volume in the list is multiplied with this scaling factor before the mean and covariance is computed.

−i level
: is the report level, either 'silent', 'info' or 'std'.

LIMITATIONS

All volumes have to have same size. Input file format can be VAPET, XPRIME, ANALYZE. All volio input data types are accepted, i.e. byte, ushort, sshort, float, double.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

FLIP() FLIP()

NAME

flip - flip a volume along X-, Y- or Z-axis convert_vol - change datatype, scale intensity, compute min/max

SYNOPSIS

flip [-l] [-u] [-t] inputfile [outputfile]

DESCRIPTION

Flips (mirrors) a volume along an axis.

OPTIONS

-l Flip image left/right (x)

-u Flip image up/down (y)

-t Flip image top/bottom (z) If no outputfile is specified, the inputfile will be overwritten.

LIMITATIONS

Input file format can be VAPET, XPRIME, ANALYZE. All volio input data types are accepted, i.e. byte, ushort, sshort, float, double.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1)

GAUSS_VOL() GAUSS_VOL()

NAME

gauss_vol − apply Gaussian smoothing kernel to a volume

SYNOPSIS

gauss_vol [−c | −p] [−f | −s] sx sy sz inputfile [outputfile]

DESCRIPTION

Applies a Gaussian filter to the volume. The filter size is specified in the 3 dimensions by sx, sy and sz.

OPTIONS

sx sy sz
Size of Gaussian kernel. What unit this is measured in is determined below.

−c Filter size is represented in units of cm

−p Filter size is represented in units of pixels (default)

−f Filter size is full width half max (fwhm) (default)

−s Filter size is standard deviation

−g cutoff
Cutoff size for filter. Measured in fwhm's. Default is 4.

File is converted to float for filtering and then converted back to original datatype. If no outputfile is specified, the inputfile will be overwritten.

LIMITATIONS

Input file format can be VAPET, XPRIME, ANALYZE. All volio input data types are accepted, i.e. byte, ushort, sshort, float, double, although the volume is converted to float internally during filtering.

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

NAME

warp_reconcile - reconcile a2b and b2a warp fields

SYNOPSIS

warp_reconcile [**-m**] [**-s** *stopcrit*] [**-n** *maxiterations*] [**-i** *siglevel*] **field_a2b field_b2a** [reconciled_a2b [reconciled_b2a]]

DESCRIPTION

This function reconciles two displacement (output of **compute_warp**) fields which are ideally each others inverse. Each field a2b and b2a represent a non-linear warp template-to-subject and subject-to-template. The reconciled displacement fields are modifications of these two so that the transformations a2b(b2a) and b2a(a2b) are neutral when sampling at the displacement vector origins. Note that this operation is not necessary to perform if compute_warp produced a "reconciled warp".

Note that the displacement fields a2b and b2a don't need to be the same resolution, all they need is to cover the same field of view.

The program reconciles the displacement fields iteratively, and prints out the consistency of the reconciled fields.

OPTIONS

-m means measure consistency and exit right away.

-s stopcrit
sets the stopping criteria (min avr. vector move, def 0.0005).

-n maxiterations
sets the maximum number of iterations.(def. 100)

AUTHOR

Ulrik Kjems <uk@imm.dtu.dk>

SEE ALSO

gauss_vol(1), **histo_vols**(1), **subsample**(1), **file2vapet**(1), **file2ana**(1), **file2xprime**(1), **headerinfo**(1), **resize_vol**(1), **convert_vol**(1), **covar**(1), **flip**(1), **region**(1), **compute_warp**(1), **warp_reslice**(1), **warp_reconcile**(1),

Bibliography

[1] Anil K. Jain, *Fundamentals of Digital Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1989.

[2] P. Toft, *The Radon Transform - theory and applications*, Ph.D. thesis, Department of Mathematical Modelling, Technical University of Denmark, 1993–1997.

[3] J. Nolte, *The Human Brain*, Mosby Year Book, 1993.

[4] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, Apr. 1975.

[5] R. P. Woods and S. R. Cherry, "Mri-pet registration with automated algorithm.," *Journal of Computer Assisted Tomography*, vol. 17(4), pp. 536–546, 1993.

[6] R. P. Woods, S. R. Cherry, and J. C. Mazziotta, "Rapid automated algorithm for aligning and reslicing pet images.," *Journal of Computer Assisted Tomography*, vol. 16, pp. 620–633, 1992.

[7] John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1996.

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[9] P. H. Sørensen, "Master thesis: Deconvolution models for fMRI," M.S. thesis, Department of Mathematical Modelling, Technical University of Denmark, 1997.

[10] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, Massachusetts, Aug. 1994.

[11] P. Churchland and T. J. Sejnowski, *The Computational Brain*, MIT Press, Cambridge, Massachusetts, London, 1992.

[12] Radford M. Neal, "Probabilistic inference using markov chain monte carlo," Tech. Rep. CRG-TR-93-1, Department of Computer Science, University of Toronto, Sept. 1993.

[13] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," in *IEEE Computer Vision and Pattern Recognition*, 1983, pp. 448–453.

[14] G. E. Hinton, "Deterministic boltzmann learning performs steepest descent in weight space," *Neural Computation*, vol. 1, pp. 143–150, 1989.

[15] L. K. Hansen, L. N. Andersen, U. Kjems, and J. Larsen, "Revisiting boltzmann learning: Parameter estimation in markov random fields," in *IEEE Conference on Acoustics Speech and Signal Processing, Atlanta GA*, 1996, vol. 6, pp. 3395–8.

[16] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified em algorithm," *IEEE Transactions on Medical Imaging*, vol. 9, pp. 84–93, 1990.

[17] D. M. Higdon, J. E. Bowsher, V. E. Johnson, T. G. Turkington, D. R. Gilland, and R. J. Jaszczak, "Fully bayesian estimation of gibbs hyperparameters for emission computed tomography data," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 516–526, Oct. 1997.

[18] S. Geeman and D. Geeman, "Stochastic relaxation, gibbs distributions, and the baysian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Pattern Analysis and Machine Intelligence, Nov. 1984.

[19] V. E. Johnson, W. H. Wong andf X. Hu, and C.-T. Chen, "Image restoration using gibbs priors: Boundary modeling, treatment of blurring, and selection of hyperparameter," *PAMI*, pp. 413–425, 1991.

[20] X. Ouyang, W.H. Wong, V.E. Johnson, X.and Hu, and C.-T. Chen, "Incorporation of correlated structural images in pet image reconstruction," *IEEE Transactions on Medical Imaging*, pp. 627–640, 1994.

[21] M. Ohlsson, C. Peterson, and B. Söderberg, "Neural networks for optimization problems with inequality constraints - the knapsack problem," *Neural Computation*, vol. 5, pp. 331–339, 1993.

[22] M. Lagerholm, C. Peterson, and B.Söderberg, "Airline crew scheduling using potts mean field techniques," *submitted to Operations Research*, 1997.

[23] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, " " *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[24] J. Zerubia and R. Chellappa, "Mean field annealing using compund gauss-markov random fields for edge detection and image estimation," *IEEE Transactions of Neural Networks*, vol. 4, July 1993.

[25] Peter Alshede Philipsen, Ulrik Kjems, Peter Toft, , and Lars Kai Hansen, "Restoring functional pet images using anatomical mr images," in *Interdisciplinary Inversion Workshop 5*, Sept. 1997.

[26] A. Blake, "Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 2–12, 1989.

[27] D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph.D. thesis, California Institute of Technology, 1991.

[28] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1992.

[29] J. E. Bowsher, V. E. Johnson, T. G. Turkington, R. J. Jaszczak, C. E. Floyd, and R. E. Coleman, "Fully bayesian estimation of gibbs hyperparameters for emission computed tomography data," *IEEE Transactions on Medical Imaging*, vol. 16, no. 5, pp. 516–526, Oct. 1997.

[30] C. Peterson and J.R. Anderson, "A mean field learning algorithm for neural networks," *Complex Systems*, vol. 1, pp. 995–1019, 1987.

[31] M. W. Pedersen and D. G. Stork, "Second order methods in boltzmann learning: An application to speech reading," in *NIPS*, 1997.

[32] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[33] S. C. Strother, J. R. Anderson, X. L. Xu XL, J. S. Liow, D. C. Bonar, and D. A. Rottenberg, "Quantitative comparisons of image registration techniques based on high-resolution mri of the brain.," *Journal of Computer Assisted Tomograpy*, vol. 18, pp. 954–962, 1994.

[34] R. P. Woods, Scott T. Grafton, Colin J. Holmes, Simon R. Cherry, and John C. Mazziotta, "Automated image registration: I. general methods and intrasubject, intramodality validation," *Journal of Computer Assisted Tomography*, 1997, in press.

[35] R. P. Woods, Scott T. Grafton, John D.G. Watson, Nancy L. Sicotte, and John C. Mazziotta, "Ii. intersubject validation of linear and nonlinear models," *Journal of Computer Assisted Tomography*, 1997, in press.

[36] J. Talairach and P. Tournoux, *Co-planar Stereotactic Atlas of the Human Brain: 3-Dimensional Proportional System. An approach to Cerebral Imaging*, Georg Thieme Verlag, 1988.

[37] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall, 1982.

[38] Y. Kosugi, M. Sase, H. Kuwatani, N. Kinoshita, T. Momose, J. Nishikawa, and T. Watanabe, "Neural network mapping for nonlinear stereotactic normalization of brain mr images," *Journal of Computer Assisted Tomography*, vol. 17(3), pp. 455–460, 1993.

[39] D. L. Collins, P. Neelin, T. M Peters, and A. C. Evans, "Automatic 3d intersubject registration of mr volumetric data in standardized talairach space," *Journal of Computer Assisted Tomography*, vol. 18(2), pp. 192–205, 1994.

[40] D. L. Collins, T. M Peters, and A. C. Evans, "An automated 3d non-linear image deformation procedure for determination of gross morphometric variability in human brain," *Proceedings of Conference on Visualization in Biomedical Computing*, vol. SPIE Vol. 2359, pp. 180–190, 1994.

[41] M. E. Alexander and R. L. Somorjai, "The registration of mr images using multiscale robust methods," *Magnetic Resonance Imaging*, vol. 14, no. 5, pp. 453–468, 1996.

[42] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Computer Vision and Graphics Image Processing*, vol. 46, pp. 1–21, 1989.

[43] A. C. Evans, D. L. Collins, and C. J. Holmes, "Automatic 3d regional mri segmentation and statistical probability anatomy maps," *In press*, 1995.

[44] U. Grenander, M. I. Miller, and G. E. Christensen, "Deformable anatomical data bases using global shape models," in *1992 EIHB Workshop*, 1992.

[45] S. Minoshima, R. A. Koeppe, K. A. Frey, and D. E. Kuhl, "Anatomic standardization - linear scaling and nonlinear warping of functional brain images," *Journal of Nuclear Medicine*, vol. 35(9), pp. 1528–1537, Sept. 1994.

[46] Y. Amit, "A non-linear variational problem for image matching," *Siam Journal of Scientific Computation*, vol. 15, no. 1, January 1994.

[47] J. Boes, B. Kim, K. A. Frey, and C. Meyer, "Mri-pet head registration via iterative minimization of mutual information," in *Proceedings of the 2nd Int.Conf. on Functional Mapping of the Human Brain.* June 1997, p. 149, Academic Press.

[48] B. Kim, J. Boes, K. A. Frey, and C. Meyer, "Mutual information for automated unwarping of rat brain autoradiographs," *Neuroimage*, vol. 5(1), pp. 31–40, Jan. 1997.

[49] C. Studholme, D. L. G. Hill, J. Wong anf M. N. Maisey, and D. J. Hawkes, "Registration measures for automated 3d alignment of pet and intensity distorted mr images," in *Proceedings in Image Fusion and Shape Variability Techniques.* July 1996, pp. 186–193, IEEE Computer Society Press ISBN 0-8186-7367-2.

[50] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Incorporating connected region labelling into automated image registration using mutual information," in *Proceedings of the IEEE workshop on Mathematical Methods in Biomedical Image Analysis, San Fransisco CA.* June 1996, pp. 23–31, IEEE Computer Society Press ISBN 0-8186-7367-2.

[51] C. Studholme, D. L. G. Hill, and D. J. Hawkes, "Automated three-dimensional registration of magnetic resonance and positiron emission tomography brain images by multiresolution optimisation of voxel similarity measures," *Medical Physics*, vol. 24, pp. 24–35, 1997.

[52] S. Lakshmanan and H. Derin, "Gaussian markov random fields at multiple resolutions," *Markov Random Fields, Theory and Application*, pp. 131–157, 1993.

[53] G. E. Christensen, M. I. Miller, and M. Vannier, "A deformable magnetic resonance textbook bason on elasticity," in *Application of Computer Vision in Medical Image Processing.* March 1994, Stanford University.

[54] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "A deformable neuroanatomy based on viscous fluid mechanics," in *Proc. of the 1993 Conference on Information Sciences and Systems*, Mar. 1993, pp. 211–216.

[55] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "3d brain mapping using a deformable neuroanatomy," *Physics in Medicine and Biology*, vol. 39, pp. 609–618, 1994.

[56] G. E. Christensen, R. D. Rabitt, and M. M. Miller, "Deformable templates using large deformation kinematics," *IEEE Transactions of Image Processing*, vol. 5(10), pp. 1435–1447, 1996.

[57] M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander, "Mathematical textbook of deformable neuroanatomies," in *Proc. of the National Academy of Science*, December 1993, vol. 90:24, pp. 11944–11948.

[58] K. Friston, J. Ashburner, C. D. Frith, J-B. Poline, J. D. Heather, and R. S. J. Frackowiak, "Spatial registration and normalization of images," *Human Brain Mapping*, vol. 2, pp. 165–189, 1995.

[59] D. L. Collins, G. Le Goualher, R. Venugopal, A. Caramanos, A. C. Evans, and C. Barillot, "Cortical constraints for non-linear cortical registration," *Lecture Notes in Computer Science, Visualization in Biomedical Comptuting*, pp. 307–316, 1996.

[60] I. Law, C. Svarer, and O. B. Paulson, "Characterization of the cortical responses during the performance of reflexive and antisaccadic eye movements," *Human Brain Mapping*, p. 1:323, 1995.

[61] T. R. DeGrado, T. G. Turkington, J. J. Williams, C. W. Stearns, J. M. Hoffman, and R. E. Coleman, "Performance characteristics of a whole-body pet scanner," *Journal of Nuclear Medicine*, vol. 35, pp. 1398–406, 1994.

[62] J. R Moeller and S. C. Strother, "A regional covariance approach to the analysis of functional patterns in positron emission tomographic data," *Journal of Cerebral Blod Flow Metabolism*, vol. 11, pp. A125–135, 1991.

[63] J. R Moeller, S. C. Strother, J. J. Sidtis, and D. A. Rottenberg, "Scaled subprofile model: A statistical approach to the analysis of functional activation patterns in positron emmision tomographic data," *Journal of Cerebral Blod Flow Metabolism*, vol. 7(5), pp. 649–658, 1987.

[64] S. C. Strother, J. R. Anderson, K. A. Schaper, J. J. Sidtis, J. S. Liow, R. P. Woods, and D. A. Rottenberg, "Principal component analysis and the scaled subprofile model compared to intersubject averaging and statistical parametric mapping: I. "Functional connectivity" of the human motor system studied with [15-o]water pet," *Journal of Cerebral Blood Flow and Metabolism*, vol. 15, pp. 738–753, 1995.

[65] K. V. Mardia and J. T. Kent, *Multivariate Analysis*, Academic Press, 1979.

[66] S. C. Strother, J. R. Anderson, K. A. Schaper, J. S. Sidtis, and D. A. Rottenberg, "Linear models of orthogonal subspaces and activated networks from functional activation pet studies of the human brain," in *Computational Imaging and Vision, Information Processing in Medical Imaging*. 1995, vol. 3, pp. 299–310, Kluwer Academic Pubs.

[67] S. C. Strother, J. R. Anderson, J. J. Sidtis, K. A. Schaper, L. K. Hansen, J. S. Liow, and D. A. Rottenberg, "Functional image registration metrics based on orthogonal subspaces," *Mapping and Modeling of the Human Brain*, 1997.

[68] B. Lautrup, L. K. Hansen, I. Law, N. Mørch, C. Svarer, and S. Strother, "Massive weight sharing: A cure for extremely ill-posed problems," in *Proceedings of Workshop on Supercomputing in Brain Research: From Tomography to Neural Networks: From tomography to neural networks, HLRZ, KFA Jülich, Germany*, H. J. Hermann, D. E. Wolf, and E. P. Pöppel, Eds. Nov. 1995, pp. 137–148, World Scientific.

[69] S. C. Strother, N. Lange, R. L. Savoy, J. R. Anderson, J. J. Sidtis, L. K. Hansen, P. A. Bandettini, K. O'Craven, M. Rezza, B. R. Rosen, and D. A. Rottenberg, "Multidimensional state-spaces for fmri and pet activation studies," in *Neuroimage 3(Part 2) 2nd Int. Conf. Functional Mapping of the Human Brain*, June 1996, p. S98.

[70] S. C. Strother, N. Lange, J. R. Anderson, K. A. Schaper, K. Rehm, L. K. Hansen, and D. A. Rottenberg, "Activation pattern reproducibility: Measuring the effects of group size and data analysis models," *Human Brain Mapping*, vol. 5, pp. 312–316, 1997.

[71] N. Mø rch, L. K. Hansen, S. C. Strother, C. Svarer, D. A. Rottenberg, B. Lautrup, R. Savoy, and O. B. Paulson, "Nonlinear versus linear models in functional neuroimaging: Learning curves and generalization crossover," in *Lecture Notes in Computer Science 1230: Information Processing in Medical Imaging*, J. Duncan and G. Gindi, Eds. 1997, pp. 259–270, Springer-Verlag.

Index