3D Coding of MR Images and Estimation of Hemodynamic Response Estimation from fMRI Data

A Thesis

Submitted for the Degree of

Doctor of Philosophy

in the Faculty of Engineering

by

R. Srikanth

Department of Electrical Engineering
Indian Institute of Science
Bangalore – 560 012

OCTOBER 2004

Acknowledgments

I acknoledge........

Abstract

This thesis mainly deals with two problems in medical image processing, viz., 3D magnetic resonance image (MRI) compression and modeling of functional magnetic resonance image (fMRI) data. In the first part of the thesis, two schemes for lossless compression of 2D and 3D MR images have been proposed. We have proposed a nonuniform mesh based interframe coding scheme for 3D MR images. The mesh has been formed in such a way that only the brain part of the image is meshed. The mesh has been designed by choosing node points along the edges and the density of the mesh elements in a region has been decided based on the optical flow between two consecutive MR slices. A large mesh element will be chosen if the optical flow in a region is small; otherwise a smaller element is chosen. A triangular mesh has been designed using delaunay triangulation. This nonuniform mesh has found to be superior to uniform mesh based scheme, wherein image is tiled by uniform mesh elements, with respect to coding performance. Affine transformation is then used to predict the next slice based on the current slice. In the textured region, where the motion compensation is reliable, we predict the current pixel using affine transformation. In regions where edges are prominent, the motion compensation using affine transformation is unreliable. This is because of unreliable motion estimation at edges due to aperture problem. In these regions, we use gradient adjusted prediction as in Context-based Adaptive Lossless Image Coder (CALIC), the sate of the art lossless image coder. We then formulate 3D energy and texture contexts as in CALIC to encode decorrelated pixels. This method, when we have applied on 3D saggital brain images, has found to be superior to uniform mesh based scheme, CALIC and the recent 3D wavelet-based schemes, like Multi Layered Zero Coding (MLZC) and 3D Embedded

Zerotree Wavelet (EZW) coders.

We have also proposed a 2D and 3D wavelet-based coding schemes with region of interest, progressive transmission and lossless coding capabilities. These functionalities are very useful in telemedicine applications. 2D and 3D separable integer wavelet transforms are respectively used to decorrelate 2D and 3D images. Correlation within the subbands is exploited using fixed size lattices. The performance of coding schemes has been further improved by incorporating context based coding for the significance and sign maps. 2D and 3D contexts are constructed using a mutual information based context quantization algorithm. The achieved lossless compression performance of the 3D scheme is comparable to the above proposed mesh-based schemes and the recent MLZC and 3D EZW coders.

The second part of the thesis deals with modeling of fMRI data for cognitive studies of brain. We have mainly dealt with the estimation of hemodynamic response function (HRF) which is useful not only for detecting the activated regions for a specific task but also to study the dynamics of the brain activation. With the introduction of event-related design paradigm, the modeling of HRF has become a research interest on its own. For example, the estimation of time to peak of HRF helps in time sequencing the activated regions. i.e., one can find which are the primary regions that are activated first for a given stimulus and which are the other associated regions of the brain that get activated later. The amplitude of response gives the strength of activation. We have modeled HRF by both parametric and nonparametric models. We have used linear model wherein the BOLD response is modeled as convolution of HRF and design paradigm. We have proposed a new probability model for fMRI time-series using probabilistic principal component analysis. Using this model, the correlated noise (base-line trends etc) in fMRI time-series is represented in terms of principal components of the fMRI data. This probability model is then used to estimate the hemodynamic response function. We have modeled HRF by both parametric and non-parametric (smooth FIR filter) models. The number of principal components to be included in the model has been determined by Akaike Information Criterion. A new bootstrap method to calculate covariance matrix has also been given to take care of spatial noise variation. We have also studied the influence of sampling rate (TR) on the accuracy of HRF estimation on both simulated and real fMRI noise data. Smooth FIR filter model has been found to be a good method at high sampling rates and parametric model, a better choice at lower sampling rates.

We have also developed a classification and estimation algorithm using which voxels having similar HRFs have been classified together. Classification is performed based on the assumption that neighbouring voxels will have same class labels. This spatial smoothness has been imposed by modeling class labels as Markov Random Fields (MRF). Gibbsian distribution has been used as a prior for class labels. Expectation maximization algorithm is then used to determine class labels and HRF functions. PPCA has been used to model the fMRI voxel time series. Classification and estimation accuracy have been verified for different sampling rates and HRF models. This algorithm is useful in analyzing event-related fMRI studies.

The assumption in PPCA model for fMRI time series is that the time series is stationary. Recent studies on null fMRI data have revealed that fMRI time series exibhit $1/f$-like spectral properties which are characteristics of nonstationary signals. Such signals can be conveniently characterized by fractional brownian motion (fBm). Recently, fBm models for fMRI time series have been successfuly used for finding activated regions in the brain. We have also used fBm based models for the estimation of HRF. HRF is again modeled using parametric and nonparametric models as mentioned above. The probability models have been built on the assumption that wavelet transforms with sufficient number of vanishing moments decorrelate fBm processes. The baseline drifts have also been estimated along with HRF. The obtained estimation accuracy for both gaussian and smooth FIR filter models is comparable to that obatained by PPCA model. We have also compared the PPCA and fBm methods with the exsisting method in literature which uses polynomial bases for modeling baseline drifts and smooth FIR filter model for HRF. Both PPCA and fBm models using smooth FIR filter model for HRF have been found to be superior to this method. However, the fBm model for nonstationay fMRI time series has not shown much improvement over the stationary PPCA model.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image and signal processing tools are now being widely used for medical images. Automatic processing and analysis of medical images are required in number of areas like acquisition, image construction, enhancement, compression, segmentation and registration. Medical images can be acquired by number of modalities like X rays, computer aided tomography, magnetic resonance imaging (MRI), positron emission tomography (PET) and ultrasound imaging. Image acquired by each one of these modalities provide different information about the organ imaged. They also possess different characteristics and hence warrant different processing methodologies. The invention of MRI has revolutionized the filed of medical imaging. Apart from being noninvasive, MRI can image human soft tissue anatomy. It has several advantages over other imaging modalities enabling it to provide three-dimensional (3D) data with high contrast between soft tissues. MRI is also now being widely used for functional studies of brain. The use of MRI for functional studies of brain is termed as functional magnetic resonance imaging (fMRI). fMRI can provide better time and spatial resolutions compared to PET which is a traditional tool for functional studies.

1.1 Motivation

We carried out this research based on the clinical needs of our colloborators, National Institute of Medical Health and Neuro Science (NIMHANS), Bangalore, India. In a typical medical institute like NIMHANS, huge amount of medical image data is generated on a daily basis. This data needs to be stored for future study and follow up. This requires a large amount of storage space which is especially true with three-dimensional (3-D) medical data obtained by the use of three dimensional (3D) imaging modalities like magnetic resonance imaging (MRI), computerized tomography (CT) and positron emission tomography (PET). With the recent devolopments in telemedicine, there is also a need for faster transmission of images over the communication lines. Hence, compression of medical images plays an important role for efficient storage and transmission. Hence efficient compression of 3D medical data with lossless reconstruction functionality is one of the requirements of our colloborators.

fMRI is now being widely used for clinical purposes in NIMHANS. Functional aspects of brain are to be studied before and after surgery of a brain part. fMRI studies can be used to assess the damaged caused by surgery on the functional aspect of that brain portion. For these studies, the freely available statistical parameter mapping (SPM) software is used to find the activated regions of brain for a given task. One important requirement of NIMHANS is to study the dynamics of brain activation. This study can reveal how an activation is propogated for a given task in an healthy and abnormal subject. These kind of studies are not possible with SPM package. One way of studying the dynamics of response is to model hemodynamic response function (HRF).

In this thesis, we deal with the above problems in medical image processing:

- Compression of 3D MR Images.
- Estimation of Hemodynamic Response Function (HRF) from fMRI data.

We have developed two schemes for 3D MRI compression. The emphasis is on lossless compression which is needed for compressing image data before diagnosis.

In the second part of the thesis, we address the problem of estimating HRF function from fMRI data. The emphasis is on modeling fMRI data which contains low frequency drifts and random noise. We have developed two probability models for fMRI. The first model assumes fMRI time-series to be stationary. In the second model, we have modeled fMRI time-series as a non-stationary process. We have used both parametric and non-parametric models for HRF.

In the following, we give a basic principle of MRI, issues in medical image compression, a review of medical image compression literature. In the latter part of this chapter, we discuss the basic principle of fMRI and review different models for fMRI data. We finally end with organization of the thesis.

1.2 Principle of MRI

Magnetic Resonance Imaging is based on the phenomenon of nuclear magnetic resonance (NMR). MRI is normally based on using the NMR signal from the hydrogen nuclei in water molecules, which have an intrinsic "spin" and hence a magnetic moment. In the presence of a magnetic field, the nuclear spins "precess" around the direction of the field at a frequency proportional to the strength of the field. The spins can lie either parallel or antiparallel to the field. The difference in energy between the two states is very small, and those nuclei that are parallel to the field have a slightly lower energy. The Boltzmann equation therefore predicts that the number of spins parallel to the field will be only slightly greater than the number pointing in the opposite direction. Since the spins precess around the magnetic field with random phases, there will, in a normal, unperturbed state of a sample, be a very small net magnetization aligned parallel to the applied field. This unperturbed longitudinal magnetization is denoted as M_o.

The NMR signal is detected by applying a pulse of radio frequency waves that is typically a few milliseconds in duration. This rotates the direction of M_o away from the direction of magnetic field direction. However, the spins are continually interacting with the varying magnetic fields arising from the motion of surrounding molecules. Once the radio frequency pulse is switched off, M_o returns to its unperturbed state over a characteristic time known as the T_1 relaxation time. This time, which depends on the local nature of a sample, is one of the main sources of "contrast" in clinical MRI.

Excitation also brings the spins into phase, so that the net magnetization now precess around the direction of the magnetic field in the transverse plane. This precession can be detected by electromagnetic induction in a pickup coil. The amplitude of the signal is proportional to the size of M_o before the radio frequency pulse was applied. If, however, the magnetic field is fluctuating in time, for example due to molecular motion, then the spins will precess at slightly different frequencies. This leads to "spin dephasing", causes a loss of coherence and makes the signal decay with time. This process takes place over a characteristic time known as T_2, which is another major source of contrast in MRI.

Spin dephasing can also be caused by spatial distortions in the magnetic field within a sample. These distortions might arise, for example, from variations in magnetic susceptibility between different tissues. This dephasing arises from static magnetic fields that vary with position in the sample but not with time. The time constant with which the magnetization decays, when the effects of both the static and the time-varying fields are included, is known as T_2^*.

1.3 Need for and issues in Compression

In this chapter, we discuss the need for compression of medical images (both volumetric (3D) and 2D slices) and review various compression schemes reported in the literature.

Recent advances in medical imaging technology have led to ubiquitous use of medical images for diagnosis. As a result, huge amount of medical image data is generated on a daily basis. This data needs to be stored for future study and follow up. This requires a large amount of storage space which is especially true with three-dimensional (3-D) medical data obtained by the use of three dimensional (3D) imaging modalities like magnetic resonance imaging (MRI), computerized tomography (CT) and positron emission tomography (PET). With the recent devolopments in telemedicine, there is also a need for faster transmission of images over the communication lines. Hence, compression of medical images plays an important role for efficient storage and transmission. Compression schemes can be broadly divided into two categories viz., (1) Lossless and (2) Lossy. In the lossless schemes, the images can be reconstructed exactly after compression. This exact reconstruction is achieved at the expense of low compression ratios, of the order of 2:1 to 4:1. On the other hand, lossy compression schemes achieve higher compression ratios of the order of 30:1, but exact reconstruction of images is not possible. There will be some loss of information in lossy compression schemes. In the medical imaging scenario, there is concern that the diagnosis should not be affected by the loss of information due to compression schemes. Also, medical images are subjected to further processing like segmentation, enhancement and filtering whose performance should not be affected by lossy compression schemes. For example, operations like enhancement and filtering may result in accentuating the degradations caused by lossy compression. Also, there are legal issues which do not permit lossy compression schemes [2]. However, region-based lossless schemes may be used once the diagnosis is completed. Regions of importance can be compressed without any loss and other regions can be compressed with lossy schemes. With these hybrid schemes, one can achieve high compression ratios. In telemedicine applications, where images are to be transmitted to remote locations, regions of interest can be transmitted prior to other regions without any loss. Other regions are transmitted by compressing them with lossy schemes. This results in effective transmission of medical images over communication channels. Standard compression schemes like JPEG2000 provide such functionalities. Hence, in the medical imaging scenario, there is a need for efficient lossless and lossy coding schemes. In this work, the main focus is on lossless schemes for 3D (volumetric) medical images. We develop two algorithms: one based on nonuniform mesh and spatial transformations and the other based on integer wavelet transforms with progressive transmission capability. In the next section, we give a review of various schemes reported in the literature. We divide this review into two parts. In the first part, we discuss about the spatial domain schemes and in the second part, we discuss various wavelet-based schemes.

1.4 Review of Medical Image Compression Schemes

1.4.1 Spatial domain schemes

The lossless (reversible) compression schemes broadly consist of two steps: (1) decorrelation and (2) entropy coding. In the decorrelation step, the redundancy in the image due to interpixel dependencies is reduced. In the coding step, the decorrelated pixels are encoded using either Huffman or arithmetic coding. Below, we discuss various decorrelation techniques used for 2D and 3D medical images. Roos et al studied various reversible decorrelation schemes for 2D and 3D medical images [3] to [4]. In [3], various reversible decorrelation schemes for 2D medical images were studied. They compared predictive decorrelation schemes such as Differential Pulse Code Modulation (DPCM), Adaptive DPCM (ADPCM) with reversible transform-based scehmes such as Discrete Cosine Transform (DCT), subband coding and Walsh-Hadamard and with multiresolution schemes like Laplacian Pyramid and Hierarchical Interpolation (HINT). They found that HINT outperforms all other schemes. In [5], Roos et al investigated schemes for compressing time sequences of medical images and they compared these schemes with HINT. They tried to use correlation across the sequences by using motion compensation techniques like block motion estimation and local motion estimation. These schemes are evaluated by applying them on sequences of coronary X-ray angiograms, ventricle angiograms and liver scintigrams. They conclude that estimation of interframe motion is not advantageous for image compression and interframe compression yields entropies comparable to intraframe (2D) HINT at higher computational costs. In [4], Roos et al studied reversible decorrelation schemes for 3D medical images. In view of the results obtained in [3], two methods are considered, viz., DPCM and HINT. They extended DPCM and HINT to 3D images in order to exploit the correlation within and across the slices. They applied these methods on 3D CT, MR and SPECT images. They conclude that the increase in efficiency obtained by extending the 2D methods to 3D is small.

An alternative approach to account for correlation across slices in 3D images is by ”motion” compensation schemes. This term denotes the ”deformation” between two slices, since there is no motion across slices in 3D images. Since most of the schemes for 3D medical image compression were adapted from video coding schemes, the term "motion" is interchageably used for "deformation". Block matching algorithm (BMA) [6] is a commonly used tool for motion compensation in video coding.

Lee et al [7] proposed a lossy compression scheme based on motion compensation (using BMA) for 3D medical images. In this method, the current slice is predicted based on previous slice and the residue (difference between the predicted and actual slice) is transformed by DCT. The DCT coefficients are then quantized. The bit allocation schemes and quantizers are designed based on the statistics of the DCT coefficients of residue images. BMA assumes only translation motion between two consecutive slices. A close look at volumetric images indicates that the assumption of translation motion is too simplistic. The variations between two successive slices are too complex to be approximated by translation motion. There are more sophisticated algorithms in video coding literature which account for complex deformations. These methods are based on spatial transformations [8].

In [9], Aria Nosratinia et al used the scheme in [8] for interframe coding of 3D MR images. In this scheme, the slice to be coded is partitioned by uniform mesh (triangular or rectangular) elements. These mesh elements are deformed in the previous slice by determining the motion of node points. An affine transformation is determined between the regular element in the current slice and the deformed mesh element in the previous slice. This spatial transformation models not only translation motion but also rotation and scaling. Motion compensation is achieved by determining the corresponding coordinates of pixels in each mesh element in the previous image. Unlike block-matching algorithms, spatial transformation based models generate a continuous, piecewise linear motion filelds. The motion compensated images are then coded using a zero-tree wavelet based coder. Although wavelet based scheme is used for coding, we classified this scheme in spatial domain schemes since the motion compensation is achieved in the spatial domain.

Uniform mesh elements do not adequately model the deformation because pixels inside a single mesh element may contain multiple motions. A better approach is to design a mesh depending upon the content of the image and motion across the slices [10]. This approach leads to a nonuniform mesh where the size of mesh element at each location depends upon the smoothness of the location and the motion of its pixels. Another advantage of this method is that only the object of interest is meshed. Medical images, in particular MR images, consist of noisy background and clinically useful foreground. The foreground forms only $40-60\%$ of the entire image. The background is only noise and does not contain any useful clinical information. If uniform mesh based schemes are used, then one needs to unnecessarily code the background information and spend a lot of bit budget. On the other hand, content based mesh schemes code only the useful foreground region and ignore background region. In our work, we take this approach to code the 3D MR images. The details of the scheme are given in chapter 2. We use context-based entropy schemes in conjunction with content-based mesh scheme to devise a region of interest lossless scheme. We now review context-based entropy schemes for lossless compression of images.

So far, we have discussed various decorrelation schemes employed for medical images. Over the years, there is also a significant progress in the entropy coding of data using contexts or conditioning events. In [11], Rissanen and Lagdon considered entropy coding as two parts: (1) source modeling and (2) coding. In conventional Huffman or arithmetic coding, the implicit source model is a memoryless model wherein, the symbols to be coded are assumed to be independent of each other. The lower bound on the average bits per symbol that can be achieved is given by the entropy of the symbols. The assumption of memoryless model is very conservative. They showed that by building source models which predict a sequence with higher probability, one can achieve better compression compared to memoryless models. The source models are built using contexts or conditioning events which exploit the interdepedencies of source symbols. Here, we discuss briefly about the context models and then review the various schemes reported in the literature using context-based models.

Let α be a sequence of length N to be coded. Suppose the sequence has J distinct symbols denoted by the set $A = \{a_1, a_2, ..., a_J\}$. A is called as alphabet of the sequence.

Let the probabilities of occurrence of each symbol be $\{P(a_1), P(a_2), ..., P(a_J)\}$. Assuming memoryless model (i.e., occurrence of each symbol is independent of other), then the lower bound on the number of bits per symbol, required to represent the sequence is given by the entropy $H(Y)$ for the ensemble Y with sample space A:

$$H(X) = -\sum_{j=1}^{J} P(a_j) log_2 P(a_j) \tag{1.1}$$

Huffman and Arithmetic coders can reach close to this bound if the data sequence is large enough. Hence, the amount of compression one can achieve depends on the source model. If the symbols in the sequence have interdependencies (memory) then the memoryless source model is not adequate. A better source model can reduce the lower bound on the number of bits per symbol. For sequences with memory, source models can be built using contexts which are defined as functions of symbols already coded.

Consider the sequence described above with an alphabet size of J. Let there be K contexts denoted by $C = \{c_1, c_2, ..., c_K\}$. The contexts are defined based on a predetermined rule, which is a function of symbols already coded. Let the sequence α to be coded be denoted as

$$\alpha : y(1)y(2)y(3)....y(i)....y(N-1)$$

where, $y(i) \in A, for \quad i = 1..N$. Let the sequence of contexts β corresponding to α be denoted as:

$$\beta : x(1)x(2)x(3)....x(i)....x(N-1)$$

where, $x(i) = f(y(1)y(2)....y(i-1))$, $x(i) \in C, for \quad i = 1..N$ and $f(.)$ is a predetermined rule for defining contexts. Let $P(a_j|c_k)$ be the conditional probability of symbol a_j given the context c_k for $j = 1..., J, \quad k = 1..., K$. Let $P(a_j, c_k)$ be the joint probability. Let X and Y be the two discrete ensembles with respective sample spaces C and A. Then the conditional entropy of Y given X is given by:

$$H(Y|X) = --\sum_{j=1}^{J}\sum_{k=1}^{K} P(a_j, c_k) log_2 P(a_j|c_k) \tag{1.2}$$

where, $H(Y|X)$ is the conditional entropy of Y given X. This quantity represents the lower bound on the bits per symbol required to code the sequence α when the context sequence β is given. It can be shown that $H(Y|X) \leq H(X)$. The improvement $H(Y) - H(Y|X) \geq 0$ is due to the mutual information provided by the contexts about the source symbols. Hence, an improvement over the memoryless model in terms of average bits required for the symbols to be coded.

In [12], Rambadran and Keshi Chen used context-based source models in conjunction with decorrelation schemes like DPCM, WHT and HINT for reversible compression of medical images. They found that context-based coding enhances the performance of each scheme. The contexts are initially built using gradients along orthogonal directions in the original (or transformed) image. The contexts are then adaptively selected using context splitting procedure based on the estimated value of the pixel. Arithmetic coder is used for entropy coding. They found that DPCM along with context-based coding performs better for MR and ultrasound images and WHT with context-based coding works well for X-Ray images. These schemes are proposed for 2D images.

In [13], Midtvik and Hivig proposed a lossless compression scheme for MR images. This is also a context-based model similar to the scheme proposed in [12]. They use static source models in the sense that contexts are not changed unlike in [12]. They also observe that the statistical properties of background and foreground in MR images are different. The foreground region is highly correlated and therefore can be easily compressed. The background is noisy and is difficult to compress. They use separate source models for background and foreground.

Context-based, adaptive, lossless image codec (CALIC) [14] is a state-of-the-art method for lossless compression of 2-D images. This scheme uses a nonlinear gradient adjusted predictor (GAP) for image decorrelation. It uses a combination of texture and energy contexts to adapt the nonlinear predictor to varying image statistics. Energy contexts are used to encode the prediction errors.

The above context-based schemes are designed for 2D images. In our work, we have used the context-based entropy coding schemes on 3D MR images. The 3D MR images are decorrelated using both inter and intraframe correlation. The interframe correlation is exploited using motion compensation and intraframe correlation is exploited using gradient adjusted prediction. The motion compensation fails in regions where edges are predominant due to aperture problem. In the texture region, where the motion compensation is reliable, we predict the current pixel using spatial transformation. In edges where motion compensation is unreliable we use Gap Adjusted Prediction (GAP) as in [14]. We then formulate 3D energy and texture contexts as in [14] to encode decorrelated pixels. The details of the scheme are given in the next chapter. To our knowledge, this is the first time a context-based scheme is used for interframe, lossless compression using content-based mesh. This is one of the main contributions of the thesis. We also apply this scheme for uniform mesh-based scheme proposed by [9] and compare with the above content-based mesh scheme.

1.4.2 Wavelet-based Schemes

Wavelet based schemes, apart from giving good compression performance, can provide important functionalities like progressive transmission and region of interest (ROI) coding which are very useful for teleradiology applications. 2D wavelet-based schemes can also be extended to 3D schemes for compressing volumetric image data. 3D wavelet transforms decorrelate images both within and across slices. The current 2D image compression standard JPEG2000 uses wavelet transform and provides the above desired functionalities. But 3D compression is out of the scope of JPEG2000. In this section, we review various wavelet-based schemes for 2D and 3D images reported in the literature.

After the introduction of Embedded Zero Tree Wavelet (EZW) [15] by Shapiro, the use of wavelet transform (WT) has gained popularity for still image and video compression. EZW algorithm gives an embedded bit stream for progressive transmission of images. EZW exploits the correlation of wavelet coefficients across the subbands by using a data structure called zero-tree. This algorithm is based on the assumption that if a wavelet coefficient at a certain scale is insignificant with respective to a threshold, then all the coefficients of the same orientation in the spatial location at finer scales are also insignificant with respect to that threshold. If this assumption is true, then all these coefficients can be represented by a single symbol. In [15], Shapiro showed that the probability of zero-tree assumption is very high and hence significant compression can be achieved. Said and Pearlman [16] modified this algorithm and introduced a coding scheme popularly known as Set Partitioning in Hierarchical Trees (SPIHT). SPIHT is an alternative implementation of EZW and they showed that it is faster and outperforms EZW. Layered Zero Coding (LZC) [17] and Embedded Block Coding with Optimized Truncation of the embedded bit-streams (EBCOT) [18] are the other popular wavelet based coding schemes.

Conventional wavelet transforms map integer-valued image data to real-valued wavelet coefficients due to which lossless compression of images is not possible. Hence for lossless compression, it is necessary to have a transformation which maps integer-valued data to integer-valued coefficients. In [19], [20], Sweldens et al., showed that such a transformation can be obtained using a method called lifting scheme. This tranformation, which gives integer-valued coefficients, is called Integer Wavelet Transform (IWT). IWT can be used in place of WT in EZW and SPIHT for lossless compression of 2D images. The current still image compression standard JPEG2000, which is a wavelet based compression scheme, provides an option for lossless compression using IWT. SPIHT and EZW have also been extended to 3D image and video compression [21], [1] using separable 3D WT or IWT transforms.

Schemes like EBCOT [18], JPEG2000 (which is based on EBCOT) and [22] which exploit intraband redundancies rather than interband redundancies (using zero-tree). These algorithms use either fixed size blocks or quadtree decomposition or combination of both in wavelet domain to represent significance maps. In [22], Munteanu et al proposed a lossless compression scheme for coronory angiographic images. It uses integer wavelet transform for decorrelating the image and outputs embedded bit stream for progressive transmission. They also show theoretically that the number of symbols needed to code zero regions with a fixed size block based method is lower than the number of zero-tree symbols. Results suggest that this scheme outperforms EZW and SPIHT for coronory angiographic images. This scheme can also be easily extended for region of interest coding.

In [23], Hugues Benoit-Cattin et al proposed a lossy 3D wavlet-based coding scheme for 3D X-ray images which have good resolution in all directions (356 μm). 3D separable WT is used to decorrelate the volume data both within and across slices. Uniform scalar quantization and 3D lattice vector quantization are used in wavelet domain. They also proposed a distortion minimization algorithm for selecting the best number of decompositions and the best set of qunantizers in order to minimize the overall mean square error. They reported performances based on SNR and subjective evaluation by radiologists. However, for 3D CT and MR images, spatial resolution is less than 1 mm and resolution in other direction can vary from 1 mm to 10 mm. In [24], Jun Wang and H.K.Huang proposed a similar algorithm for 3D MR and CT images. They used different wavelets for intra and inter slice directions. They observed that Haar wavelets perform better when the slice thickness is greater than 3 mm. They also observed that the performance of 3D scheme will be better than the 2D scheme only when slice thickness is small. This is because the slices are more correlated at smaller slice thickness. These schemes do not have progressive transmission and lossless reconstruction functionalities.

Recently, a number of wavelet based schemes, [1],[25], [?], [26] for volumetric medical image data have appeared in the literature. These schemes provide useful functionalities like progressive transmission and region of interest coding.

In [1], Gloria Menegaz and Jean-Philippe Thiran proposed an object-based coding of MR images with a progressive transmission capability. They provided a provision to code regions of interest without any loss and rest of the regions are coded with loss. This enables the diagnostically relevant regions to be represented without any loss. They used 3D integer wavelet transform to decorrelate the 3D MR data. They modified 3D EZW and LZC algorithms for the region of interest coding. Their results suggest that both the coders have comparable performances.

One main issue with integer wavelet transforms is that they may not be unitary. Therefore, the quantization error in the wavelet domain will not be equal to the mean square error in the spatial domain. Hence lossy compression performance will not be optimal for nonunitary transforms. In [25], Xiong et al addressed this problem. They proposed a general 3D integer wavelet packet transform structure that allows implicit bit shifting of wavelet coefficients to approximate a 3D unitary transformation. They then applied modified versions of 3D SPIHT and 3D embedded subband coding with optimal truncation (ESCOT) schemes on 3D medical data and found that these algorithms achieve a very good performance both in terms of lossy and lossless compression over other schemes.

In [26], Schelkens et al proposed three wavelet-based compression schemes for volumetric medical image sets. They incorporated functionalities like progressive transmission with lossless compression option and resolution scalability. They also designed a new 3D DCT based coding scheme which is used as a benchmark to compare the three wavelet based coders. They argue that exploiting intraband correlation is more fruitful than exploiting interband correlation using zero-trees. Their first two coding schemes are based on octtree based coding of the significance map. The third scheme is a 3D extension of EBCOT. In all the schemes, they use context-based adaptive arithmetic coding to code the bitstreams. They also observe that the 3D schemes are sensitive to slice thickness. 2D schemes like JPEG2000 work better when for large slice thickness because the interslice correlation decreases with the increase in slice thickness. The 3D versions of EBCOT schemes in [25] and [26] differ with respect to context selection.

In our work, we give wavelet based 2D and 3D compression schemes capable of lossless compression with a progressive transmission capability. These schemes exploit intraband redundancies instead of interband redundancy. They use fixed size blocks (4×4 for 2D and $4 \times 4 \times 2$ for 3D) for coding the significance map as against quadtree representation in [26]. We also use context based arithmetic coding to code the bit streams generated by the schemes. Usually, one can expect that as number of contexts increases, the coding performance is better. But in practice, a large number of contexts results in context dilution. Since the number of symbols to be coded is finite, there will not be many number of symbols or no symbols in some contexts. Hence the estimation of probabilities of symbols in those contexts for arithmetic coding is not reliable and results in poor coding performance. Hence, selection of contexts for optimal performance is still a research problem. The selection of optimal contexts from the possible contexts is called context quantization. EBCOT provides contexts for coding zero, sign and refinement bits. 2D schemes can be easily extended to 3D. But selection of proper contexts for 3D schemes is not trivial. Many algorithms for context quantization are available in the literature. For example, [27], [28] and [29] use linear projection to project high dimensional contexts to low dimensional space and then use either dynamic programming or Lloyd-Max quantization to get required number of contexts. These algorithms are sub-optimal in the sense that information is lost during projection. Xiong et al use these kind of schemes for selecting 3D contexts in [25]. In our work, we use a mutual information based context selection proposed by [30]. This algorithm finds the required number of contexts from the possible number of contexts using a merging technique. The contexts are merged which result in minimal reduction in mutual information. This algorithm is used for context selection for both 2D and 3D schemes. The details of the scheme are given in chapter 3.

1.5 What is fMRI?

MRI is a noninvasive scanning technique which is capable of producing anatomical images of soft tissue in the body by exploiting the magnetic properties of hydrogen nuclei, contrary to X-ray imaging. MRI can also be used for measuring a range of other tissue-specific parameters. Functional Magnetic Resonance Imaging (fMRI) is one example of this. fMRI has been widely used for neurofunctional studies of the brain since 1992. fMRI has a better spatial and temporal resolution than the Positron Emission Tomography (PET) apart from being noninvasive. fMRI uses different magnetic properties of oxy- and deoxyhaemoglobin to visualize localized changes in blood flow, blood volume and blood oxygenation in the brain. These are in turn indicators for local changes in neural activity. By exposing a subject to controlled stimuli, which are carefully designed to affect only certain brain functions, it is possible to estimate the anatomical location of neurons involved in the corresponding functions. Brain function may then be mapped to brain anatomy by combining fMRI scans with anatomical scans obtained through conventional MRI.

Blood Oxygen Level Dependent (BOLD) signal

The brain is a complex organ that contains hundreds of billions of basic processing units, called neurons, connected to thousands of others. Neurons are specialized nerve cells that process and transmit information by releasing chemical messengers called neurotransmitters and firing electric signals along nerve fibers. These processes require a lot of energy, which is provided by the oxidation of glucose that is supplied to cells by blood flowing through the brain. Neural activity therefore raises the consumption of glucose and oxygen in the part of the brain where activity is taking place. This triggers a change in local "hemodynamics", a term that describes the blood flow and blood volume.

One effect of neuronal activity is to make blood flow more quickly through brain tissue. This increase in cerebral blood flow (CBF) overcompensates for the higher consumption of oxygen and actually raises the local blood oxygenation. This change in the blood oxygenation due to neuronal activity alters the magnetic resonance signal in a measurable way. This signal known as BOLD is the basis of fMRI. This effect is very sensitive to brain activity and is relatively simple to measure.

Measurement of BOLD

The BOLD effect can be attributed to the properties of hemoglobin, the iron-containing molecule that is found in red blood cells and used by the body to transport oxygen. When hemoglobin combines with oxygen it forms oxyhaemoglobin, which, like water and brain tissue, is weakly diamagnetic. However, deoxyhaemoglobin, the form of the molecule without oxygen is paramagnetic. The presence of deoxyhaemoglobin in red blood cells means that the magnetic susceptibility of deoxyhaemoglobin blood is different from that of brain tissue.

The change in the BOLD signal is small, just a few percent of the overall MRI signal when measured at the magnetic strength of 1.5 T (tesla) that is typical of most clinical scanners. Recently 3 T machines are used to increase the number of spins aligned along the field direction and to accentuate the effects of the differences in susceptibility. Both effects strengthen the BOLD signal and make it easier to measure, although repeated measurements are still required to obtain reliable maps of brain activation.

In the presence of a strong applied magnetic field such as that used in an MRI scanner, these differences in the magnetic susceptibility of blood and tissue set up spatially varying magnetic fields at the vessel boundaries. These varying magnetic fields further increase the rate of decay of the signal. The rate of this T_2^* decay depends on the concentration of oxygen in the blood as well as the volume of blood. When brain activity occurs, it causes more blood to flow to active brain tissue, which overcompensates for the increased oxygen consumption and lowers the amount of deoxyhaemoglobin. The signal therefore decays at a slower rate and its amplitude in active brain regions increases very slightly. It is this effect that forms the basis of BOLD fMRI.

Design Paradigms

The fMRI experiment is carried out using two design paradigms: (1) Block design and (2) Event-related or single-trial design. In the block design paradigm, the subject is asked to repeat the task for extended periods of time (blocks of, say 30 seconds), interleaved with similarly long periods of inactivity. In this design paradigm, the focus of research has been in improving the detection methods for identifying regions of activation. On the other hand, in the event-related or single-trial design paradigm, the task is performed once, followed by a period of inactivity before the next trial [37]. This allows the full response evoked by the task to be analyzed. In this design paradigm, the interest is not only in identifying the regions of activation but also to delineate different kinds of activation elicited due to "events" introduced in the experiment. This kind of analysis requires the characterization of Hemodynamic Response Function (HRF). With the introduction of event-related design paradigm, the modeling of HRF has become a research interest on its own. Reliable detection of activated regions also needs the characterization of HRF. It also helps in studying the dynamics of the activation. For example, the estimation of time to peak of the response helps in time sequencing the activated regions. i.e., one can find which regions are activated first for a given stimulus. The amplitude of response gives the strength of activation. Estimation of HRF is the main focus of this work. We model HRF by both parametric and nonparametric models. The proposed algorithms can be used for both block and event related design paradigms. We also propose an algorithm which can be used to classify regions having different HRFs which could have been elicited by events in the event related design paradigm. In the next section, we review different methodologies used so far in the literature for fMRI data analysis.

1.6 Review of fMRI data models

A typical fMRI voxel time series can be broadly considered to have three components namely,

1. Experimentally induced BOLD response, if the voxel belongs to the activated region.

2. Structured noise in the data corrupting the BOLD response. This noise is due to physiological sources such as respiratory or cardiac cycles and head movement. Also, noise is introduced by long-term instabilities of the scanner baseline. Other cognitive processes unrelated to the experimental task may contribute to physiological noise. These noise sources appear as ”trends” and ”fluctuations” in the time series.

3. Random Noise due to scanner.

Since the signal intensity changes only by $2-5\%$ due to BOLD response, the noise components need to be accounted to achieve a reliable detection of activated regions. One way of accounting for the noise components is to remove these components by bandpass filtering the data and the other way is to incorporate these components using appropriate models in the data model. In this section, we review different methodologies and models used so far, for processing fMRI voxel time series.

1.6.1 Hemodynamic Response Function Models

In the earlier works for block periodic stimulation experiments, a periodic model for BOLD response is used. The simplest possible model for BOLD response is a periodic boxcar function with a value of 0 during baseline and a value of 1 during the stimulation. The model is refined by shifting the boxcar function $4-8$ seconds to account for the delay in the response. In [38] and [39], a sine wave model with the same period as the stimulation is proposed for HRF modeling. This resembles the observed response more than the boxcar function and can be used to interpret frequency analysis of time series wherein the power at stimulation frequency is the parameter of interest. The cross correlation between the sinusoidal reference function and the time series is used as a test statistic for detecting activated regions. This model is further improved in [40] and [41] by modeling the response function as a linear combination of cosines and sines at the stimulation frequency and their first and second harmonics. This corresponds to a Fourier bases for the response function.

In their seminal work, Boynton et al., [42] showed that the BOLD response can be represented by a linear model. This model is less empirical since they attempt to explain the response from a biological viewpoint, and thus aim at a better understanding of the underlying processes. This in turn allows for generalizations to non-periodic experiments and other more detailed studies of the brain. The observed BOLD response at m^{th} voxel ($y_m(t)$) can be represented using the linear model as:

$$y_m(t) = \sum_{k=0}^{N} x(t-k)h_m(k) \tag{1.3}$$

where $x(t)$ is the stimulation function (a boxcar function with value 1 during activation and 0 otherwise) and $h_m(t)$, the transfer function of the linear system, is interpreted as the response function to a short stimulation of one time-point. The assumptions underlying this model are that the response is time invariant (stationary), in the sense that the impulse response $h_m(t)$ is the same for all stimulation time points, and that the impulse responses combine additively over prolonged periods of stimulation. Hence time series at each voxel m can be represented as a linear system response to a stimulation x. In this work, we call the system response $h_m(t)$ as HRF, and the convolution of system response with the stimulation function as BOLD response.

Several models, both parameteric and nonparameteric, have been proposed in the literature. Among the parametric models, the discrete poisson function [43], the gaussian function [44], [37] and the gamma function [42] are popular parametric models. The estimation of HRF boils down to estimating the parameters of the model. The assumption of parametric function for HRF may result in some bias in estimating the actual HRF. In [45], [46], [47], [48] nonparametric models for HRF are used. These models do not use any specific models for HRF and the latter is represented by a FIR filter. The filter coefficients are then estimated. Since the number of coefficients to be estimated are large, the problem becomes ill-posed. Several regularization methods are used without biasing the HRF estimation. In the event-related design paradigms, more careful modeling of HRF is required to find regions of different activations due to events. Improving detection of activated voxels has, until recently, been the primary focus in fMRI studies. With the advent of event or single trial fMRI experiments, the characterization of HRF has become a research interest in its own right. Hence efficient estimation of HRF is very important for fMRI data analysis.

1.6.2 Structured Noise Models

Filtering

As mentioned earlier, one way of removing structural noise in fMRI data is by bandpass filtering. In [49], several filtering techniques for removing structured noise components are compared. The baseline fluctuations are slow with respect to the length of an experimental trial, i.e., their frequency is lower than the stimulation frequency. Hence a lowpass filter with cutoff frequency less than stimulation frequency can be used for removing these baseline fluctuations. The performances of moving average (MA) filter, Lowpass FIR (LP) filter, autoregressive (AR) filter and Kalman filters are compared for baseline correction in [49]. Subtraction of the filter output from the input signal effectively results in highpass filtering which separates the anatomical information (ie., the time-independent part) from the functional activation. Estimation of the baseline by LP filter is found to be sufficient. A cutoff frequency of $1 - 1.5$ times the trial length (in time steps) is reported to be sufficient. But filtering introduces correlation in the time series and hence there is a trade off between removing the noise components and the cost of reducing degrees of freedom. Also, the filtering methods rely on approximate periodicity of the physiological noise; at least frequency band of this noise must be relatively small and constant over time. Due to aliasing effects, this may not be a realistic assumption.

Trend Models

An alternative approach is to incorporate the structural noise in the fMRI time series model itself. Several subspace methods have been used to model the systematic noise components. In [50], a detrending subspace of dimension eleven is used to detrend the observed time series. Detrending is achieved by projecting the time series onto the space spanned by detrending subspace and removing the latter component from the observed time series. In [51], three low frequency sine/cosine terms (six dimensions) at selected frequencies are used to form the detrending subspace which they call as confounds. In [45], polynomial functions are used as basis functions to represent the trend component. They use the following linear model to represent the observed time series:

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{P}\lambda + \mathbf{e} \tag{1.4}$$

where, $\mathbf{y}$ is the $N \times 1$ observed time series vector at a given voxel, X is the $N \times K$ convolution matrix and $\mathbf{h}$ is the unknown hemodynamic response function of dimension $K \times 1$. The columns of the matrix P span the noise subspace. Polynomials of order 2 are used as bases for the noise subspace. λ represents the coordinates of the noise component with respect to the columns of P. The noise component $\mathbf{e}$ is modeled as independent and identically distributed (iid) gaussian noise with unknown variance σ^2. The HRF $\mathbf{h}$ is modeled as a smooth FIR filter. Bayesian framework is used to estimate HRF. In [52], zero and first order polynomials are used as bases for the noise subspace.

These choices of bases are *ad hoc* in nature. There is no systematic way of choosing the bases and appropriate dimension of the noise subspace. In [41], a systematic way is suggested for choosing noise subspace. The following model is used for activation detection for block design experiments:

$$\mathbf{y} = \mathbf{A}\theta + \mathbf{B}\phi + \mathbf{e} \tag{1.5}$$

where A is a truncated Fourier series (columns span the signal subspace), θ is an unknown vector representing the coefficients for the response. B is the unknown matrix whose columns span the noise subspace, ϕ is the unknown vector representing the coordinates of the noise space and $\mathbf{e}$ is white gaussian noise with unknown variance σ^2. The unknown quantities θ, B, ϕ and σ^2 are determined by Maximum Likelihood Estimation (MLE) by imposing an additional constraint that the signal subspace is orthogonal to the noise subspace ($A^tB = 0$). They found that the columns of B are principal components of the sample correlation matrix of voxel time-series vectors with the signal component removed. Akaike information criterion is used to estimate the dimension of the noise subspace. There is no restriction on the choice of the matrix A. However, this method cannot be used for the event-related design paradigm where, the assumption of periodic response is not valid. Another shortcoming of this method is that $\mathbf{e}$ is assumed to be constant for all the voxels. But it is observed that noise characteristics change from region to region [53].

In [54], state space model for fMRI time series is proposed. The model is of the form

$$y(t) = a(t) + z(t)b(t) + e(t), \quad e(t) \sim N(0, \sigma^2) \tag{1.6}$$

where, $z(t)$ is a model for BOLD response, obtained by convolving the stimulation function with a poisson function with mean θ and lag d time points. The terms $a(t)$ and $b(t)$ are latent processes which represent trend and the magnitude of the BOLD response, respectively. Random walk models are used for these latent variables.

$$\begin{aligned} a(t) &= 2a(t-1) - a(t-2) + \zeta(t), \quad \zeta(t) \sim N(0, \sigma_\zeta^2) \\ b(t) &= 2b(t-1) - b(t-2) + \eta(t), \quad \eta(t) \sim N(0, \sigma_\eta^2) \end{aligned} \tag{1.7}$$

The parameters (d, θ) are first estimated and then expectation maximization algorithm is used to estimate the unknown variances, trends $(a(t))$ and magnitude components $(b(t))$. Although this model is very flexible in modeling the BOLD and trend components, the complexity of the model is very high which affects the inferential power of the method.

In [55], a wavelet based method is used to find the trend in the fMRI time series. The fMRI time series is assumed to be composed of three components namely, a smooth trend (θ), response to stimulation function $\mathbf{x}$ and a correlated noise component $\mathbf{e}$.

$$\mathbf{y} = \theta + \beta\mathbf{x} + \mathbf{e} \tag{1.8}$$

where β is a scalar which measures the strength of the response. The trend component is estimated in the wavelet domain. Since trend is a signal whose frequency is less than the stimulation frequency, it is assumed to be dominant in the low frequency bands of the wavelet domain. This assumption is used to estimate the trend component. The correlated noise component $\mathbf{e}$ is modeled as a long memory process. A wavelet transform with sufficient number of vanishing moments decorrelates a long memory process [56]. This property is used in the wavelet domain to estimate θ and β. A statistical t-test is used to find regions of activation.

1.6.3 Random Noise Models

After modeling HRF and structured noise, what remains to be modeled is random noise. A simple model for random noise is white gaussian noise with variance σ^2. This random noise is due to physiological sources or scanner. It is observed that even in the absence of experimental effects, fMRI time series has a serial dependence. Several models have been proposed to model this serial dependence. Generalized linear model (GLM), also called as linear regression model, is a commonly used model in the literature for modeling the fMRI time series. It is given by

$$\mathbf{y} = \mathbf{A}\beta + \mathbf{e}, \qquad \mathbf{e} \sim \mathbf{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{1.9}$$

where A is a $N \times M$ matrix, called as a design matrix whose columns consist of a model for HRF, a constant mean value term and trend terms. The noise term $\mathbf{e}$ is coloured with unknown covariance matrix Σ. In general, there are two ways in which the error term can be modeled. In the first method, called as pre-whitening method, the elements of covariance matrix are estimated [40], [57], [53]. Then the time series is whitened by multiplying both sides of the equation 1.9 by a whitening matrix K obtained by Cholesky decomposition of Σ $(KK^t = \Sigma)$.

$$K^{-1}\mathbf{y} = \mathbf{K^{-1}A}\beta + \mathbf{K^{-1}e}, \qquad \mathbf{K^{-1}e} \sim \mathbf{N}(\mathbf{0}, \mathbf{I}) \tag{1.10}$$

where I is an identity matrix. In the second method, called as pre-colouring method, the elements of Σ are predicted by convolving the data and the design matrix by a known smoothing kernel [58]. Let S be the smoothing kernel matrix. Multiplying both sides of 1.9 by S, we have

$$S\mathbf{y} = \mathbf{SA}\beta + \mathbf{Se}, \qquad \mathbf{Se} \sim \mathbf{N}(\mathbf{0}, \mathbf{S\Sigma S^t}) \tag{1.11}$$

Now $S\Sigma S^t$ is approximated as $\sigma^2 S^t$. This approximation is exact only when $\Sigma = \sigma^2 I$, i.e., the noise is white. But the idea is that by smoothing the data, the inference is less sensitive to deviations from the white noise model. Fundamentally, the two approaches are similar. However, these two approaches have relative advantages and disadvantages. Pre-whitening approach is theoretically more efficient but estimates may be biased if the unknown covariance matrix Σ is not correctly estimated. Pre-colouring is less efficient but less prone to bias provided the colouring matrix is robust enough to impose its predicted form on the autocorrelation function. Pre-colouring also has the risk of attenuating high frequency features in the data induced by experimental effects. At present, there are no sufficient results to judge the adequacy of either of these methods in dealing with event related experiments or experiments conducted at high field strength (3 Tesla).

Both these methods assume that the fMRI time series is stationary. Recently, it has been observed that fMRI time-series exhibits slow decay of autocorrelation function, characteristic of long memory processes. These processes typically have disproportionate spectral power at low frequencies, i.e., their spectrums are $1/f$-like. These processes are nonstationary in nature and cannot be modeled by the conventional AR and ARMA models meant for stationary time series. In [59], [60] and [55], fMRI time series is modeled by fractional Brownian motion (fBm). fBm is a convenient tool to model $1/f$ processes. fBm is characterized by a parameter called Hurst component H. It is observed that wavelet transform (WT) with a sufficient number of vanishing moments decorrelates $1/f$ processes [61], [62]. In [56], it is further shown that WT exhibits Karhunen-Loeve property for $1/f$ signals. This property of WT is used in [59], [60] and [55] to model fMRI time series. Multiplying WT matrix W on both sides of 1.9,

$$W\mathbf{y} = \mathbf{WA}\beta + \mathbf{We}, \qquad \mathbf{We} \sim \mathbf{N}(\mathbf{0}, \mathbf{\Lambda}) \tag{1.12}$$

where Λ is a diagonal matrix whose entries depend on the scale of the WT and the parameter H. The diagonal entries can be estimated by estimating H or by employing variance estimators in each wavelet band. In [59], a resampling method is used in wavelet domain for activation detection. The probability distribution of the test statistic under null hypothesis is estimated. A good control over Type I (false positives) is achieved compared to conventional AR models. In [60], a wavelet-based, best linear unbiased estimator (BLU) is used to estimate β. Compared to ordinary least squares and ARMA-based estimators, this method is found to be efficient and gives a good Type I error control. They conclude that wavelet based BLU is an optimal estimator of GLM with long memory errors.

Apart from these models, there are large number of spatio temporal models in the literature which combine both spatial and temporal models for fMRI time series [63], [64], [37], [52], [65], [66]. Gaussian random fields or Gibbs distributions are used to incorporate spatial information in the time series models. In this work, we are mainly concerned with temporal modeling of fMRI time series for HRF estimation. We model HRF by both parameteric and nonparametric models. We propose two fMRI time series models. In the first model, a probability model is developed, which represents fMRI noise in terms of principal components of the data using probabilistic principal component analysis (PPCA). To account for the spatial variance in noise, we propose a bootstrap method in the wavelet domain to generate multiple realizations of a given time series. This bootstrap method preserves the second order statistics of the time series. These multiple realizations are then used to estimate the covariance matrix from which principal components are computed for the probability model. Chapter 4 gives a detailed account of this method. In the second method, fBm is used to model the random noise in fMRI time series. HRF is estimated in the wavelet domain using the decorrelating property of WT for fBm. This method also estimates trends in the fMRI data. Since HRF estimation is carried out at every voxel, spatial variations in fMRI noise are taken into account. Chapter 5 gives a detailed account of this method. These two methods are compared with the method of [45], which uses polynomial bases for trends.

1.7 Organization of the Thesis

Chapter 2 gives a detailed description of our proposed mesh based scheme for 3D-MRI compression. A non-uniform mesh is designed based on the contents of the image. Current frame is compensated based on the previous frame using affine transformation. Context based entropy coding is used to efficiently code the residue images. This method is compared with the performance of existing uniform mesh based scheme, Context based Adaptive Lossless Image Compression (CALIC), a state of the art 2D lossless scheme and wavelet based schemes of [1].

Chapter 3 gives wavelet-based coding of 2D and 3D MR images. Integer wavelet transforms are used to facilitate lossless compression. These schemes provide region of interest, progressive transmission and lossless coding capabilities. These functionalities are very useful in telemedicine applications. 2D and 3D separable integer wavelet transforms are respectively used to decorrelate 2D and 3D images. Correlation within the subbands is exploited using fixed size lattices. The performance of coding schemes is further improved by incorporating context based coding for the significance and sign maps. 2D and 3D contexts are constructed using a mutual information based context quantization algorithm. The achieved lossless compression performance of the 3D scheme is comparable to the mesh based schemes proposed in chapter 2 and wavelet based algorithms of [1].

In Chapter 4, we propose a fMRI voxel time-series using Probabilistic Principal Component Analysis (PPCA). Using this model, the correlated noise (base-line trends etc.) in fMRI time-series is represented in terms of principal components of the fMRI data. This probability model is then used to estimate the Hemodynamic Response Function. We modeled HRF by both parametric and non-parametric (smooth FIR filter) models. The number of principal components to be included in the model is determined by Akaike Information Criterion. A new bootstrap method to calculate covariance matrix is also given to take care of spatial noise variation. We also studied the influence of sampling rate (TR) on the accuracy of HRF estimation on both simulated and real fMRI noise data.

We also present a classification and estimation algorithm using which voxels having similar HRFs are classified together. Classification is performed based on the assumption that neighbouring voxels will have same class labels. This spatial smoothness is imposed by modeling class labels as Markov Random Fields (MRF). Gibbsian distribution is used as prior for class labels. An expectation maximization algorithm is used to determine class labels and HRF functions. PPCA is used to model the fMRI voxel time series. Classification and estimation accuracy is verified for different sampling rates and HRF models. This algorithm is useful in analyzing event-related fMRI studies.

In chapter 5, we used fBm based models for the estimation of HRF. The assumption in PPCA model for fMRI time series is that the time series is stationary. Recent studies on null fMRI data has revealed that fMRI time series exhibit $1/f$-like spectral properties which are characteristics of non stationary signals. Such signals can be conveniently characterized by fractional brownian motion (fBm). HRF is again modeled using parametric and nonparametric models of chapter 4. The probability models are built on the assumption that wavelet transforms with sufficient number of vanishing moments decorrelates fBm process. The baseline drifts are also estimated along with HRF. The obtained estimation accuracy for both gaussian and smooth FIR filter models is comparable to that obtained by PPCA model. We also compared the PPCA and fBm methods with the existing method of [45] which uses polynomial bases for modeling baseline drifts and smooth FIR filter model for HRF.

Chapter 6 gives a comprehensive report on the contributions of the thesis and gives some directions for possible future work.

Chapter 2

Nonuniform Mesh-based Coding of MR Images

2.1 Introduction

In the last chapter, we discussed various algorithms for medical image compression. In this chapter, we propose a non-uniform mesh-based interframe coding scheme for MR sequences. The mesh is generated based upon the content of the image which ensures that multiple motions are avoided within each element. We use optic flow between two consecutive frames for mesh generation. Recently, lossless still image compression schemes such as [12], [14], [18], [29] successfully employ context-based entropy methods. By choosing proper contexts, one can model the sources efficiently and reduce the entropy of the symbols to be coded. We employ context-based entropy coding of the residues obtained after motion compensation. We propose an efficient 3-D compression scheme based on content-based non-uniform mesh and spatial transformation for motion compensation and context-based entropy coding to code the residue. We also show that the performance of the existing uniform mesh-based scheme can also be improved by incorporating context-based entropy coding.

2.2 Pre-Processing

A typical MR image consists of two parts:

1. Air part (background)

2. Flesh part (foreground)

The flesh part contains the useful clinical information which needs to be compressed without any loss. On the other hand, the air part does not contain any clinical information. It is only noise and consumes unnecessary bit budget and impairs the performance of a compression scheme. In [13], a scheme is proposed which uses two source models, one for background and the other for foreground, and an improvement in performance is reported. However, there is no need to code the air part as there is no useful information present in it. This fact has been confirmed by the neuro-radiologist we are collaborating with. Thus, in our work, we ignore the air part. We generate image masks in such a way that the flesh part is totally included and the pixel values in the air part are made zero. The rest of this section explains an image independent algorithm for mask generation. Morphological operations can be effectively used to generate image masks, which contain a value of '1' in the foreground and a value of '0' in the background. The original image is then multiplied with these masks to obtain "background noise free" images while keeping the information in the foreground intact. The algorithm for generating the mask is given below:

1. Binarize the image with a suitable threshold obtained from the image's histogram.

2. Holes may be formed within the foreground. Close these holes using morphological 'closing' operation.

3. Background may contain spurious lines. Use morphological 'erode' operation to remove these lines.

4. The above erosion operation also erodes the boundary of the foreground region. To make sure that the mask spans the entire foreground region, use morphological 'thickening' operation to thicken the boundary of the foreground region.

5. Multiply the original image with the resulting binary mask.

Figure 2.1 shows an MR image, its mask and the image obtained after multiplication with the mask. Note that this algorithm ensures that the flesh part, which is to be compressed without any loss, remains intact while the background is suppressed. In the next chapter, we use this binary mask to represent region of interest in wavelet based coding schemes.

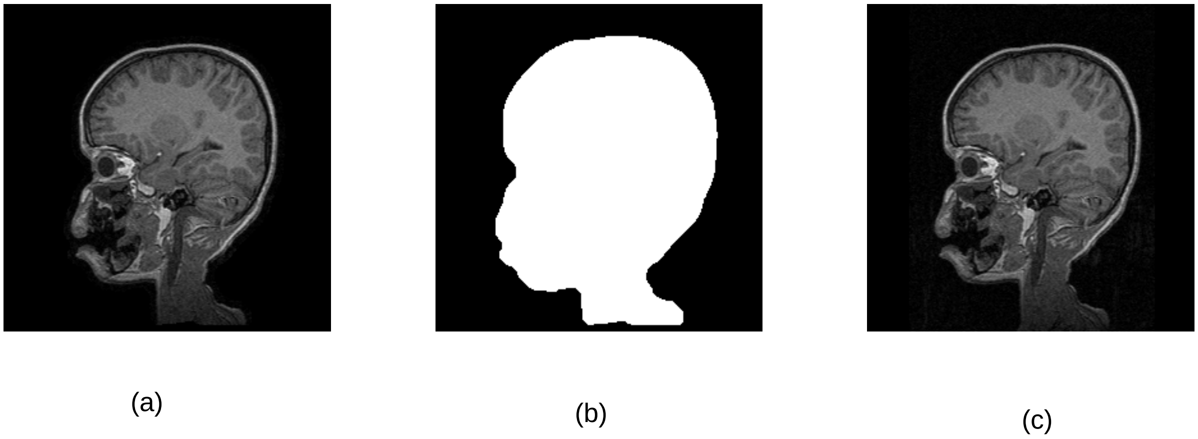

Figure 2.1: Suppression of background in an MR image using morphological operations. (a) original image (b) the generated mask (c) background suppressed image.

2.3 Content-Based Mesh Design

In mesh-based schemes, the image is divided into representative elements and the deformation of each element in the subsequent frame is modeled by a spatial transformation. In this scheme, we use triangular elements. The simplest way to divide the image is by uniform mesh elements. As mentioned earlier, this method does not adequately represent the image since each mesh element may contain multiple deformations and hence a unique spatial transformation of the element cannot model these deformations. Hence, there is a need for content-based mesh. Content-based mesh should be designed in such a way that regions with large deformations are represented by dense elements and smooth regions by few elements. Wang et al. [32] proposed a scheme based on an optimization frame work. However, most optimization schemes converge to a local optimum resulting in a mesh close to a uniform mesh. Here, we use the scheme proposed in [10]. The basic idea is to place node points in such a way that mesh boundaries align with object boundaries and that the density of the node points is proportional to the local deformations (motion). The former is based on the assumption that deformation is rigid and node points are placed along the edges. The density of mesh elements is decided from the optical flow of the region. The density is chosen to be high in regions of high optical flow and vice versa. Hence, this scheme requires the computation of optical flow between two consecutive frames. In this work, we compute the optical flow using the method of Horn-Schunck [33]. This is a gradient-based method which assumes conservation of intensity between the two images and gives a smooth optical flow as compared to Block Matching Algorithm (BMA) [31]. Let I_k be the current frame and I_{k+1} be the next frame. Mesh is to be generated on I_k by taking its spatial information and optical flow between I_k and I_{k+1}. Let $DFD(x,y)$ be the displaced frame difference which can be computed as

$$\begin{aligned} DFD(x,y) &= I_k(x,y) - \tilde{I}_k(x,y) \\ where, \quad \tilde{I}_k(x,y) &= I_{k+1}(x-u, y-v) \end{aligned}$$

where, $\tilde{I}_k(x,y)$ is an estimation of $I_k(x,y)$ based on the optical flow vector (u,v). The procedure for mesh generation is:

1. Label all pixels as "unmarked".

2. Compute the average displaced frame difference DFD_{avg} as given below:

$$DFD_{avg} = \sum_{(x,y)} \frac{DFD(x,y)^2}{K} \tag{2.1}$$

   where, K is the number of unmarked pixels.

3. Find the spatial edge map of the image using "Canny" edge detector.

4. Select a pixel as a "node" if it is "unmarked" and falls on a spatial edge and is sufficiently away from all the previously marked pixels.

5. Grow a circle about this node point until $\sum(DFD(x,y))^2$ in this circle is greater than DFD_{avg}. Label all pixels within the circle as "marked".

6. Go to step-2 until required number of nodes are selected, or the distance criterion is violated.

7. Given the selected nodes, apply a delaunay triangulation to obtain the required content-based mesh.

As mentioned earlier, the above algorithm ensures that nodes are placed along the edges and density of the elements is decided by the amount of activity between two frames which is given by the displaced frame difference DFD. We need to choose number of nodes and minimum distance between the nodes. The nodes should be selected in such a way that entire region of object is meshed. If the distance between the nodes is small and number of nodes is large, we may expect a better representation of the image. But the positions of the nodes need to be sent to the decoder as a side information. Large side information outweighs the gain obtained by smaller meshes. Hence, number of nodes and minimum distance between them need to be chosen judiciously. In this work, the minimum distance between two nodes is kept at 12 and a maximum of 200 nodes is chosen. This choice is based on our experimentation. We need to send the locations of node points to the decoder as a side information. This side information is about 0.1 *bpp*. Hence, for each frame, the mesh is to be generated based on the optical flow estimation. This may result in efficient motion compensation but this advantage may be nullified by large side information. One way to overcome this is to generate the mesh on k^{th} frame based upon the optical flow between $(k-1)^{th}$ and k^{th} frames. Here, we assume that the change in optical flow between $(k-1)^{th}$ frame to $(k+1)^{th}$ frame is small. With this procedure, the locations of node points need not be sent as a side information as $(k-1)^{th}$ and k^{th} frames are available both at the encoder and the decoder. We compare the performance of both of these methods. Figure 2.2 shows the nodes selected using the above algorithm and mesh generated using delaunay triangulation on a sample MR image.

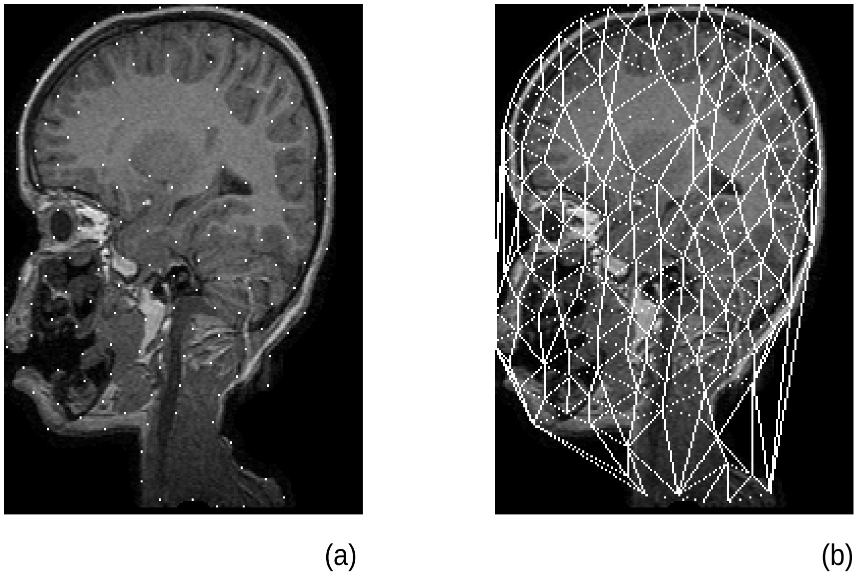

Figure 2.2: (a) Nodes selected on a sample MR image and (b) mesh generation using delaunay triangulation.

2.4 Motion Compensation using Spatial transformation

Motion compensation methods can be defined as techniques that divide images into local regions (in our case triangles) and estimate for each region a set of motion parameters. The procedure that synthesizes the predicted image of the $(k+1)^{th}$ frame $\hat{I}_{k+1}(x,y)$ from the previous frame $I_k(x,y)$ can be regarded as an image warping. This can be written as:

$$\hat{I}_{k+1}(x,y) = I_k(f(x,y), g(x,y)) \tag{2.2}$$

where, the geometric relationship between $\hat{I}_{k+1}(x,y)$ and $I_k(x,y)$ is defined by the transformation functions $f(x,y)$ and $g(x,y)$. In this work, we use affine transformations for the above geometric relationships [8]. The transformation functions for the pixels included in the i^{th} triangular element are

$$\begin{aligned} f(x,y) &= a_{i1}x + a_{i2}y + a_{i3} \\ g(x,y) &= a_{i4}x + a_{i5}y + a_{i6}. \end{aligned}$$

where a_{i1} to a_{i6} are the six deformation parameters of the i^{th} element. These parameters can be computed if the three node point correspondences of the i^{th} element in $(k+1)^{th}$ and k^{th} frames are known. These correspondences can be obtained either from the already computed optical flow or can be estimated using simple block matching algorithm. We use the latter method since motion vectors from the optical flow do not ensure the mesh connectivity. Hence, preprocessing is needed to enforce this connectivity. Instead, we employ BMA algorithm which ensures mesh connectivity. We take a 16 × 16 block with the node as the center. We assume that the maximum displacement in any direction of each node is not more than 5 pixels. We move the 16 × 16 block in the next frame within a region of 5 × 5 and take the position with minimum mean square difference as the corresponding node point in the next frame. The difference between the two positions is sent to the decoder as a side information. The above procedure is repeated for all the nodes and the triangular elements are deformed accordingly. Let (u_j, v_j), $j = 1 \; to \; 3$ be the corresponding nodes of (x_i, y_i) in the k^{th} frame. The above affine transformation can be written as:

$$\begin{pmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a_{i1} & a_{i2} & a_{i3} \\ a_{i4} & a_{i5} & a_{i6} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.3}$$

The above system of linear equations can be solved for the motion parameters a_j's, $j = 1 \; to \; 6$, for each triangular element i. The motion vectors have to be sent as side information to the decoder. We raster scan the pixels in $(k+1)^{th}$ frame and find the appropriate coordinate (u,v) in the previous frame using the corresponding mesh element's affine transformation. The coordinate (u,v) in k^{th} frame may not correspond to the grid pixel. We use bilinear interpolation to estimate the intensity $\hat{I}_{k+1}(x,y)$ at the pixel (x,y) of

$(k+1)^{th}$ frame as given below:

$$\begin{aligned}\hat{I}_{k+1}(x,y) = \lfloor (1-\alpha_1)[(1-\beta_1)\ I_k(X,Y,1) + \beta_1\ I_k(X+1,Y,1)] + \alpha_1[(1-\beta_1)\\ I_k(X,Y+1,1) + \beta_1\ I_k(X+1,Y+1,1)]\rfloor\end{aligned}$$

where, (X,Y) is the integral part and (α_1,β_1) is the fractional part of (u,v). We use the rounding operator $\lfloor . \rfloor$ so that the predicted values are integers and the residue as calculated below can be entropy coded without any loss.

$$resd(x,y) = I_{k+1}(x,y) - \hat{I}_{k+1}(x,y) \tag{2.4}$$

The pixel values in smooth regions can be predicted from the previous frame using the above affine transformation. However, the motion vectors at intensity edges cannot be calculated accurately due to the aperture problem. To overcome this problem, we estimate the pixel values at edge locations using the causal neighborhood information in the same frame as follows:

$$\begin{array}{lll} \textbf{IF} & (I_{k+1}(x,y)\ \ is\ on\ edge) & \\ \hat{I}_{k+1}(x,y) & = I_{k+1}(x-1,y) & \text{if pixel is on vertical edge} \\ \hat{I}_{k+1}(x,y) & = I_{k+1}(x,y-1) & \text{if pixel is on horizontal edge} \\ \textbf{ELSE} & & \\ Estimate & the\ intensity\ from\ affine\ transformation & \end{array} \tag{2.5}$$

where, the horizontal and vertical edges are estimated using (2.6). The thresholds, required to decide whether a pixel is on an edge or not are taken from [14]. The above estimation algorithm effectively exploits the linear correlation of neighborhood pixels. The residue can be further compressed by entropy coders like Huffman or Arithmetic coders. Recently, context-based entropy coders [11], [14] and [28] have gained prominence for lossless compression of images. If X represents the random field of the residue to be coded and if C denotes the context (or conditioning class) associated with each element of X, then one can show that the conditional entropy $H(X|C)$ will be less than or equal to the entropy $H(X)$. The reduction in entropy can be achieved by carefully choosing the contexts. In the next section, we present a practical method for choosing proper contexts for context-based entropy coding of residues.

2.5 Context-based Entropy Coding

As mentioned in the previous chapter, statistical modeling of the source being compressed plays an important role in any data compression scheme. We also discussed some schemes reported in the literature to devise contexts for entropy coding of source symbols. In [14] and [28], a practical and efficient method is given to choose contexts for lossless compression of still images. We employ a similar method in this work to choose 3D contexts. The estimation of $I_{k+1}(x,y)$ by the algorithm given in the previous section exploits only linear redundancy. It does not completely remove the statistical redundancy in the image sequence. The prediction error depends upon smoothness of the image around the predicted pixel $I_{k+1}(x,y)$ and the displacement of the pixel. To model this correlation, we formulate an error energy estimator

$$\Delta = ad_h + bd_v + cm_v + d|e_w|,$$

where $e_w = I_{k+1}(x,y) - \hat{I}_{k+1}(x,y)$ (previous prediction error) and d_h and d_v denote estimation of horizontal and vertical edges and m_v denote the displacement of the pixel (x,y). d_h and d_v are determined from the already coded pixels as follows:

$$\begin{aligned} d_h = {} & |I_{k+1}(x,y-1) - I_{k+1}(x,y-2)| + |I_{k+1}(x-1,y) - \\ & I_{k+1}(x-1,y-1)| + |I_{k+1}(x-1,y) - I_{k+1}(x-1,y+1)| \\ d_v = {} & |I_{k+1}(x-1,y) - I_{k+1}(x-2,y)| + |I_{k+1}(x,y-1) \\ & -I_{k+1}(x-1,y-1)| + |I_{k+1}(x-1,y+1) - I_{k+1}(x-2,y+1)| \end{aligned} \tag{2.6}$$

The previous error e_w is included because large errors tend to occur consecutively. The coefficients a, b, c and d are determined using least squares method. These are obtained in an off-line procedure using any two slices of our data set. The values are found to be $a = 1.27$, $b = 1.2$, $c = 0.25$ and $d = 0.02$. Now, the residue $resd(x, y)$ is conditioned on Δ so that the prediction errors are classified into classes of different variances. Since a large number of conditioning classes lead to the problem of context dilution, the number of conditioning classes need to be small. We quantize Δ into 8 levels using Lloyd-Max quantizer [34]. We found that the values of a, b, c and d vary very little for other slices in the two data sets that we used in our work. The variations in the values do not affect the quantization levels of Δ and hence the coding performance. Hence, we use the above values for all the slices.

In addition to this, we can capture higher order image patterns like texture patterns by forming additional contexts C. We form these contexts C with four causal neighbors in the current frame and four neighbours in the previous frame. The context C for the pixel (x, y) is defined as:

$$\begin{aligned} C &= [I_{k+1}(x, y-1)\ I_{k+1}(x-1, y)\ I_{k+1}(x-1, y-1)\ I_{k+1}(x-1, y+1) \\ &\quad I_k(x+1, y-1)\ I_k(x, y+1)\ I_k(x+1, y)\ I_k(x+1, y+1)]. \end{aligned}$$

We quantize C into an 8-ary binary number by using $\hat{I}_{k+1}(x, y)$ as a threshold:

$$C(k) = \begin{cases} 0, & \text{if } C(k) \geq \hat{I}_{k+1}(x, y) \\ 1, & \text{otherwise.} \end{cases}$$

As explained in [28], a context that surrounds the current pixel completely (360^o) models the image features better. Hence, we take four causal neighbors in the current frame and four causal neighbors in the previous frame which form 360^o type context. We form a compound context by combining the above context C with the 4 levels of energy context Δ. We classify the error into one of the compound contexts (α, β), where α is the energy context and β is the texture context. This can be viewed as product quantization of two independent image features. We accumulate the error in prediction in each context and maintain the number of occurrences of each context. We can assume that we make similar errors under the same context. Hence, by adding the mean of errors in each context as a bias to the earlier prediction, the prediction error further reduces. To be able to repeat this at the decoder, we calculate the mean upto the previous error. This is a feedback mechanism with one time unit delay. Let $\tilde{I}_{k+1}(x,y)$ denote the corrected prediction and is given by

$$\begin{aligned} \tilde{I}_{k+1}(x,y) &= \hat{I}_{k+1}(x,y) + \lfloor bias \rfloor \\ resdc(x,y) &= I_{k+1}(x,y) - \tilde{I}_{k+1}(x,y) \end{aligned}$$

where $bias = error(\alpha,\beta)/N(\alpha,\beta)$, where $N(\alpha,\beta)$ is the number of occurrences and $error(\alpha,\beta)$ is the accumulated error of the compound context (α,β) and $resdc(x,y)$ is the error after the improved prediction. We update the errors and counts of the context. In addition to improving the estimated value of $I_{k+1}(x,y)$, we predict the sign of the residue by using the estimated mean of the present context. The sign is predicted as follows:

IF $\{error(\alpha,\beta) < 0\}$ send $-resdc(x,y)$

ELSE send $resdc(x,y)$.

At the decoder, the reverse operation can be done by maintaining the same context errors and counts. The sign prediction helps in reducing the entropy of the residue since the uncertainty in the sign bit reduces. We classify the residue $resdc(x,y)$ into eight energy contexts as described above and use arithmetic coding in each context to further compress the residue.

2.6 Results and Discussion

We apply the schemes on 256 × 256, 8-bit MR sequences with slice thickness of 1 mm provided by National Institute of Mental Health and NeuroSciences (NIMHANS), Bangalore and on a data set used in [1] which they call as MR-MRI dataset. We call the first data set as NIMHANS dataset. These images are first preprocessed using the method suggested in section 2.2. This preprocessing step greatly enhances the performances of all the schemes. We compare the performance of the proposed schemes with the uniform mesh based scheme of [9] and CALIC, the state of the art 2-D lossless scheme [14]. We also compare our method with a 3D wavelet-based coder of [1] on MR-MRI data set. The scheme proposed in [9] is not a lossless scheme. In this scheme, a lossy zero-tree wavelet based scheme is applied to code the residue image after motion compensation. To compare the lossless performance, we encode the residue image obtained after motion compensation using a lossless entropy coder. The lossless performance of this scheme can also be improved by applying context-based entropy coding of the residual image. We also compare the performance of this scheme with our proposed schemes.

As mentioned earlier, we generate mesh in two ways. In the first method (Scheme A), mesh is generated on frame k by using the optical flow between frames k and $k+1$. This requires the node points to be sent as a side information in addition to motion vectors. In the second method (Scheme B), mesh is generated on frame k by using the optical flow between frames k and $k-1$. Since the frame $k-1$ is available at both encoder and decoder, only motion vectors of the nodes between frames k and $k+1$ need to be sent as a side information. The side information to be sent is less in this case. Hence, scheme B requires less side information to be sent but this is at the cost of mesh generation. This cost would be more if the optical flow between the frames $k-1$ and $k+1$ is more.

2.6.1 Implementation Issues in Entropy Coding

The residue symbols in each of the eight contexts can be entropy coded. Arithmetic Coding can be used since it gives flexibility in separately modeling the source and the coder. The alphabet size of residue symbols is greater than 256 as residue can take a negative or positive value. The dynamic range of residues, as suggested by our simulations, are found to be between -150 to 150. Figure 2.3 shows the histogram of symbols in each context. As it is evident from the figure, the dynamic range is much smaller in each context. For example in context 1, most of the symbols are between -8 to 8. If a practical entropy coder is to be implemented, the count of each symbol has to be at least 1. Hence, a non zero probability is to be assigned to each symbol in every context. This greatly impairs the performance of the entropy coder. To avoid this problem, histogram truncation technique has been suggested in [14] to code the symbols. In each context, they fix the alphabet size based on the statistics of the training images. However, in this work, we applied arithmetic coding on each channel without histogram truncation. In principle, histogram truncation can improve the coding performance.

2.6.2 Comparison of coding schemes

We compare the lossless coding performance of various schemes in terms of bits per pixel (*bpp*). The bit rate achieved in *bpp* is calculated as follows:

$$bpp = \frac{N_0 + N_b}{256 * 256}$$

where N_0 is number of bits required to send the side information and N_b is the number of bits required to code the symbols. In context based coding schemes, we apply arithmetic coder to each context. N_b is the total number of bits required to represent symbols of all the contexts after arithmetic coding. In the non context-based schemes, arithmetic coder is applied to the entire residue image obtained after motion compensation. Figure 2.4 shows the *bpp* values of each frame in our data base (NIMHANS data set). Clearly, the proposed schemes outperform the uniform mesh-based scheme and CALIC. The performance of uniform mesh based scheme is improved by incorporating the context based entropy coding. Table 2.1 summarizes the average performance (in *bpp*) of the above mentioned schemes. The results include the side information for motion vectors. Although the conditional entropies of the Schemes A and B are greater than that of CALIC

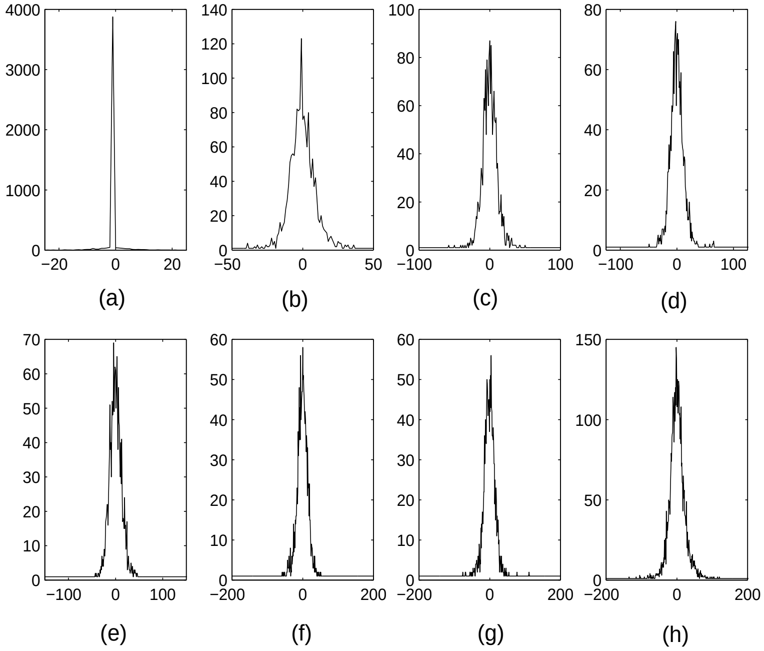

Figure 2.3: Dynamic range of residue symbols for 8 different contexts (a-h) shown as histograms.

and Uniform mesh based scheme with contexts, the number of pixels actually coded is only about 40% in the Schemes A and B. The lower conditional entropies in CALIC and Uniform mesh based scheme with contexts are due to large number of zero-valued background pixels that have to be coded. There are efficient methods available in the literature to effectively represent zero symbols. One such method is JPEG2000 standard for still image compression. To check whether applying such schemes after uniform mesh based scheme would improve the compression performance, we applied JPEG2000 scheme in lossless mode after uniform mesh based scheme. The average improvement in the bit rate is about 3.5% (2.23 *bpp* from 2.31 *bpp*), which is still higher than 2.1 and 2.0 *bpp*, respectively achieved by nonuniform schemes A and B.

To further verify the hypothesis that the nonuniform mesh based schemes better fit the consecutive slices as compared to uniform mesh based scheme, we applied uniform mesh based scheme on the flesh part of two consecutive MR images. The lossless coding performance achieved by nonuniform mesh based schemes is 0.33 *bpp* less than the uniform mesh based scheme. From this, we can conclude that nonuniform mesh based method

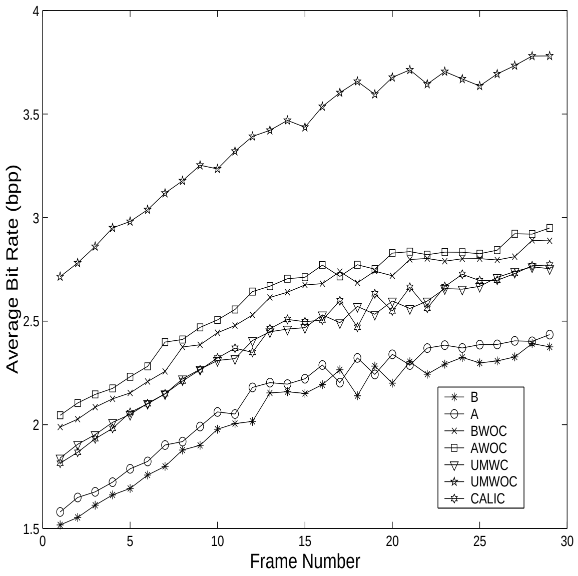

Figure 2.4: Average bit rate (*bpp*) vs frame number using different schemes on NIMHANS dataset. A: Scheme A, B: Scheme B, AWOC: Scheme A without context, BWOC: Scheme B without context, UMWOC: Uniform Mesh without context, UMWC: Uniform Mesh with context.

models the MR image slices better than uniform meshes. Figure 2.5 shows two sample consecutive MR images to be compressed. Figure 2.6 shows non-uniform mesh applied on slices 1 and 2. Figure 2.7 shows the residues after direct difference, motion compensation with uniform mesh based scheme and motion compensation with nonuniform mesh-based scheme with source modeling (scheme A). Clearly, the proposed scheme exploits the intra and inter frame correlations more effectively.

The following reasons may account for the superior performance of the proposed scheme:

1. The uniform mesh model is inadequate since each element may have multiple motions.

2. CALIC effectively exploits intraframe correlation but not interframe correlation.

3. By incorporating source models in interframe coding, both intraframe and interframe correlation is exploited.

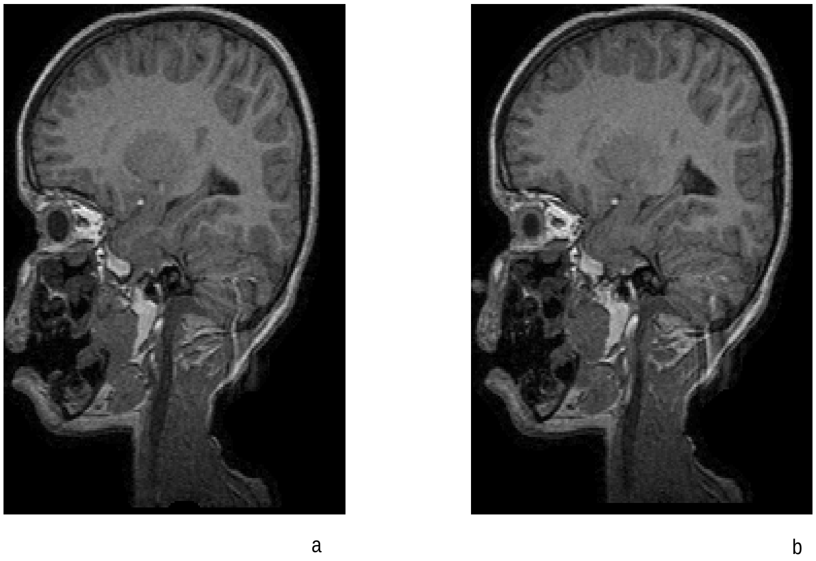

Figure 2.5: Sample consecutive Magnetic Resonance Images to be compressed, (a) Slice 1 (b) Slice 2. Slice 2 is to be compensated using Slice 1.

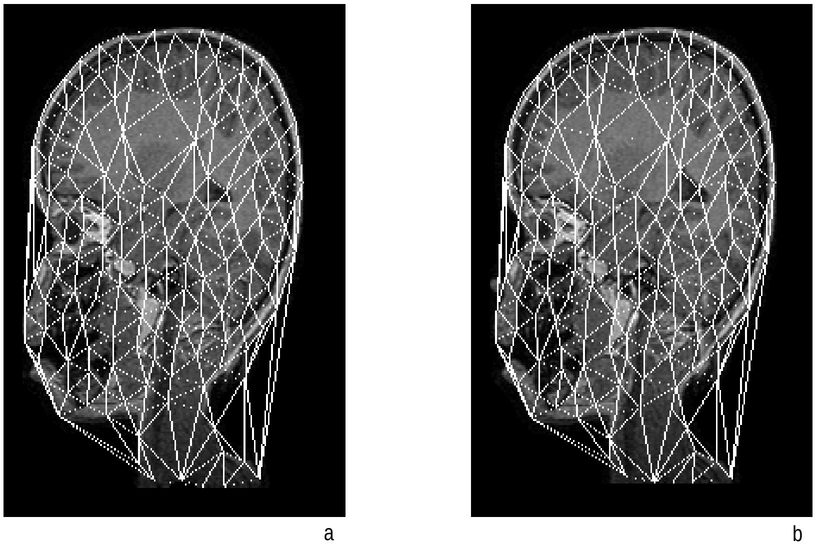

Figure 2.6: Compression using Scheme A (a) Content-based mesh on Slice 1 (b) Deformed mesh on Slice 2.

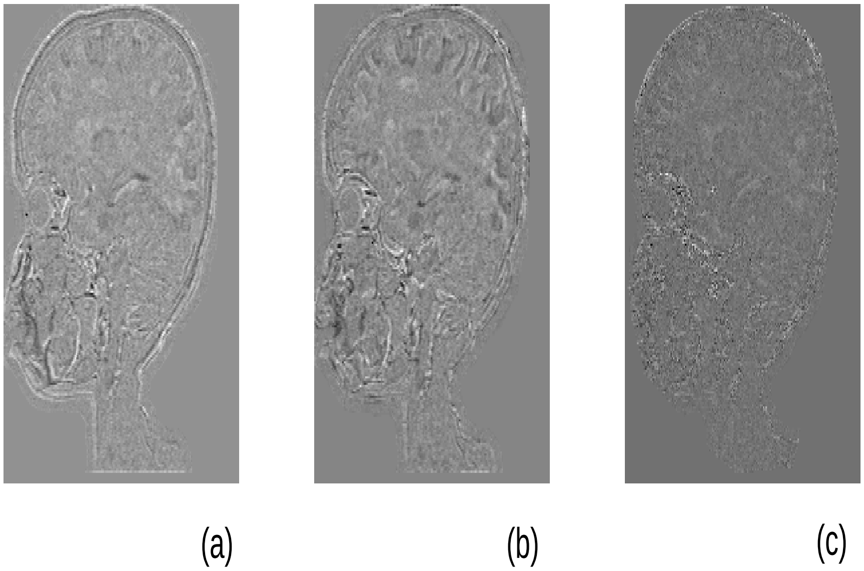

Figure 2.7: Performance comparison using residues in each method (a) direct difference between original slices 1 and 2 (b) motion compensation using uniform mesh-based scheme and (c) motion compensation by Scheme A.

Table 2.1: Performance Comparison of different schemes on NIMHANS dataset. Identical Preprocessing is used in every scheme to remove background noise. (A: Scheme A, B: Scheme B, AWOC: Scheme A without context, BWOC: Scheme B without context, UMWOC: Uniform Mesh without context, UMWC: Uniform Mesh with context, Avg: Average, SI: Side Information).

Schemes	Avg.Bit Rate(*bpp*)	Avg.Cond Entropy (*bpp*)	Avg.SI (*bpp*)	Percent Pixels Coded (foreground)
A	2.1	5.27	0.1	40
B	2.0	5.28	0.01	40
AWOC	2.55	-	0.1	40
BWOC	2.49	-	0.01	40
CALIC	2.41	4.3	-	100
UMWOC	3.51	-	0.01	100
UMWC	2.31	3.79	0.01	100

Table 2.2: Performance comparison of our schemes A and B with 3D wavelet schemes 3D EZW and MLZC on the MR-MRI dataset of [1]. (Acronyms: MLZC: Multidimensional Layered Zero Coding 3D EZW: 3D Embedded Zerotree Coding) All values are in bits per pixel (*bpp*).

Scheme A	Scheme B	3D EZW	MLZC
1.98	1.89	2.27	2.143

4. The aperture problem in optical flow estimation is avoided by resorting to estimation of pixels on intensity gradients based on neighborhood of the same plane.

5. We generate nonuniform mesh in such a way that only the object in the image is meshed and the air region is left out. This straightway improves the performance of the scheme. This kind of mesh coding can be considered as "object based coding" employed in MPEG-4 and this is achieved without any additional shape information.

We also compared our results with the 3D wavelet based coding scheme of [1]. For this, we applied our schemes (A and B) on MR-MRI data provided by the authors of [1]. They use 3D embedded wavelet zerotree coder (3D-EZW) and multidimensional layered zero coding (MLZC) for compressing MR images. Table 2.2 summarizes the coding performance of these schemes. The performance of the proposed schemes are better than MLZC and 3D EZW schemes. Recently, Zixiang Xiong et al [25] reported two 3D wavelet-based coders with lossless reconstruction capability. They report lossless bit rate of about 2 *bpp* on sagittal MR images which is comparable to our schemes and that of [1], although the data sets are different. Also, the wavelet-based scheme provides progressive transmission capability which is very useful in teleradiology applications. In the present form, our schemes do not provide this functionality. However, we can apply a lossless 2D wavelet scheme on the residues of schemes A and B to get the progressive transmission. Such strategy was used in [9] for uniform mesh based schemes.

Another important aspect of 3D compression is random access of images from the compressed 3D volume. In the present form, random access is not possible since every frame depends on the previous frame. A simple solution to this problem is to use intraframe coding (CALIC) on every n^{th} slice. This results in an increase of bit rate. We found an increase of 0.21 bpp in scheme A and 0.27 bpp in scheme B, by coding every 4^{th} slice with CALIC.

2.7 Conclusion

A new lossless content-based mesh coding scheme is proposed for MR image sequences. Context-based source modeling is used to exploit the intra and interframe correlation effectively. In this work, we have modified the contexts proposed in [14]. However, any other context-based modeling can be used in conjunction with the proposed non-uniform mesh-based scheme. A simple algorithm is also given to avoid aperture problem in optical flow estimation. The achieved lossless compression performance is better than that of uniform mesh based scheme and comparable to the recent 3D wavelet-based schemes [1].

Chapter 3

Wavelet based coding of 2D and 3D MR Images

3.1 Introduction

In the last chapter, we proposed a 3D compression algorithm which exploits both inter and intra frame correlation in the spatial domain. Another way of exploiting correlation is by applying a decorrelating transform which compacts most of the signal information in few transform coefficients. Compression can be achieved by applying quantization on these coefficients. These class of compression schemes come under the category of transform coding schemes. Commonly used transformations are Discrete Cosine Transform (DCT), Walsh Hadamard transform etc. Subband techniques are those which use filter banks to decompose a signal into different subbands. Subbands are then quantized at different levels depending upon the signal information in each band. These class of coding schemes can be classified under the category of subband coders. These coding schemes are related to each other. Among subband coding schemes, wavelet-based schemes gained popularity for both image and video compression. Wavelet-based schemes naturally render useful functionalities like progressive transmission and region of interest coding. A compression scheme with progressive transmission capability is very useful in applications like telemedicine, where data needs to be transmitted at a faster rate. With this capability, image data can be transmitted from a coarse to finer resolution. The user at the receiving end can download the image up to the resolution required and if necessary up to perfect reconstruction. This kind of scheme is useful for efficient transmission of images to a distant user. Embedded image coding using zerotrees of wavelet coefficients (EZW) by Shapiro [15] and Set Partitioning in Hierarchical Trees (SPIHT) [16] by A.Said et al., are the most popular techniques. These schemes exploit interband correlation using zerotree data structure.

The current image compression standard (JPEG2000) uses wavelet transform and provides functionalities like progressive transmission, Region of Interest (ROI) based coding and lossless compression. But 3D compression is beyond the scope of JPEG2000. In this chapter, we present a 3D wavelet based coding of MR images with progressive transmission, ROI coding and lossless compression functionalities.

The conventional wavelet transform maps integer-valued image data to real-valued wavelet coefficients because of which lossless compression is not possible. Integer wavelet transforms which map integer-valued image data to integer-valued wavelet coefficients are used for lossless compression. These integer wavelet transforms can be built using lifting scheme. In this work, we use separable 2D and 3D integer wavelet transforms to decorrelate 2D and 3D images respectively. Intraband correlation as against interband correlation, as in EZW and SPIHT algorithms, is exploited to effectively code the wavelet transformed data. We use the mask, generated in the previous chapter to suppress the noisy background, to code only the required portion of the image. We also employ a context-based entropy coder to effectively code zero and sign bits of the wavelet coefficients.

In following sections, we briefly discuss integer wavelet transforms followed by 2D and 3D wavelet based coding scheme and an algorithm for context selection based on mutual information. We compare our results with the mesh based schemes proposed in the previous chapter and coding scheme of [1] and 3D EZW algorithm.

3.2 Integer Wavelet Transform

We need transforms which map integer-valued image data to integer-valued transform coefficients for compression schemes with lossless compression capability. Such schemes are essential for medical images. In [19], [20], a simple way of constructing wavelet transforms mapping integer-integer values using lifting schemes. This scheme also takes care of boundary extensions at the border of the signal and also reduces the computational complexity with respect to classical filter band implementation of [35].

In [20], it was shown that any discrete wavelet transform or two band filtering scheme with finite filters can be decomposed into a finite sequence of simple filtering steps called as lifting steps. This decomposition corresponds to a factorization of the polyphase matrix of the wavelet or subband filters into elementary matrices. These lifting steps can be obtained using Euclidean factorization of the polyphase matrix of the filter bank into a sequence of 2×2 upper and lower triangular matrices. A brief description of this lifting scheme is given in the appendix and the details can be obtained from [20].

Below, different biorthogonal wavelets used for our compression schemes are given. The filters are represented as $(N, \tilde{N})$, where, $N($ respectively $\tilde{N})$ is the number of vanishing moments of $\tilde{g}$ (respectively g). Let $x(n)$, $s(n)$ and $d(n)$ respectively represent the original signal to be decomposed, approximate coefficients and detailed coefficients of the signal $x(n)$. Let $s_o(n)$ and $d_o(n)$ be defined as

$$\begin{aligned} s_o(n) &= x(2n) \\ d_o(n) &= x(2n+1) \end{aligned}$$

$$Filter \quad (2,2)$$

$$\begin{aligned} d(n) &= d_o(n) - |(s_o(n) + s_o(n+1))/2 + 1/2| \\ s(n) &= s_o(n) + |(d(n-1) + d(n))/4 + 1/2| \end{aligned}$$

$$Filter \quad (4,2)$$

$$\begin{aligned} d(n) &= d_o(n) - |9/16(s_o(n) + s_o(n+1)) - 1/16(s_o(n-1) + s_o(n+2)) + 1/2| \\ s(n) &= s_o(n) + |(d(n-1) + d(n))/4 + 1/2| \end{aligned}$$

$$Filter \quad (4,4)$$

$$\begin{aligned} d(n) &= d_o(n) - |9/16(s_o(n) + s_o(n+1)) - 1/16(s_o(n-1) + s_o(n+2)) + 1/2| \\ s(n) &= s_o(n) + |9/32(d(n-1) + d(n)) - 1/32(d(n-2) + d(n+1)) + 1/2| \end{aligned}$$

$$Filter \quad (6,2)$$

$$\begin{aligned} d(n) &= d_o(n) - |75/128(s_o(n) + s_o(n+1)) - 25/256(s_o(n-1) + s_o(n+2)) \\ &+ 3/256(s_o(n-2) + s_o(n+3)) + 1/2| \\ s(n) &= s_o(n) + |(d(n-1) + d(n))/4 + 1/2| \end{aligned}$$

$$Filter \quad (2+2,2)$$

$$\begin{aligned} d_1(n) &= d_o(n) - |(s_o(n) + s_o(n+1))/2 + 1/2| \\ s(n) &= s_o(n) + |(d_1(n-1) + d_1(n))/4 + 1/2| \\ d(n) &= d_1(n) - |1/16(-s(n-1) + s(n) + s(n+1) - s(n+2)) + 1/2| \end{aligned}$$

$$Filter \quad (9,7)$$

$$\begin{aligned} d_1(n) &= d_o(n) - |203/128(s_o(n) + s_o(n+1)) + 1/2| \\ s_1(n) &= s_o(n) + |217/4096(d_1(n-1) + d_1(n)) + 1/2| \\ d(n) &= d_1(n) + |113/128(s_1(n) + s_1(n+1)) + 1/2| \\ s(n) &= s_1(n) + |1817/4096(d_1(n-1) + d_1(n)) + 1/2| \end{aligned}$$

where $|.|$ is a floor operation. The block diagram for 1D transform is shown in figure 3.1.

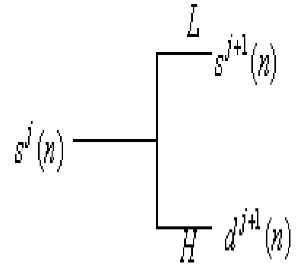

Figure 3.1: Block Diagram for 1D wavelet transform. Superscript j indicates level. L and H, respectively indicate lowpass and highpass subands.

3.2.1 2D and 3D Integer Wavelet Transforms

We need 2-D transforms for images which can be easily obtained by applying 1-D transform on rows and columns of the image sequentially. Such transforms are called separable transforms. Likewise 3-D wavelet transform can be obtained by applying 1-D transform to the three dimensions sequentially. Figure 3.1, 3.2 and 3.3, respectively, show the block diagrams of 1D, 2D and 3D wavelet transforms. Figure 3.4 shows sample MR slices to be compressed and figure 3.5 shows 3-D Separable integer wavelet decomposition of the above MR Images with 2 levels in spatial and 2 levels in temporal (across slices) domain.

3.3 Coding Scheme

3.3.1 Basic principle

The basic principle of the wavelet based coding schemes like EZW, SPIHT, LZC and EBCOT is bit plane coding in the wavelet domain, where the wavelet coefficients are encoded bitplane by bitplane starting from the most significant bit. This results in an embedded bit stream. A general bitplane coding algorithm can be explained as follows.

The scheme consists of scanning wavelet coefficients in a predefined scanning sequence and comparing the coefficients with successively smaller thresholds t_n. Let the initial threshold t_0 be defined as

$$t_0 = 2^{|log_2 w_{max}|+1} \tag{3.1}$$

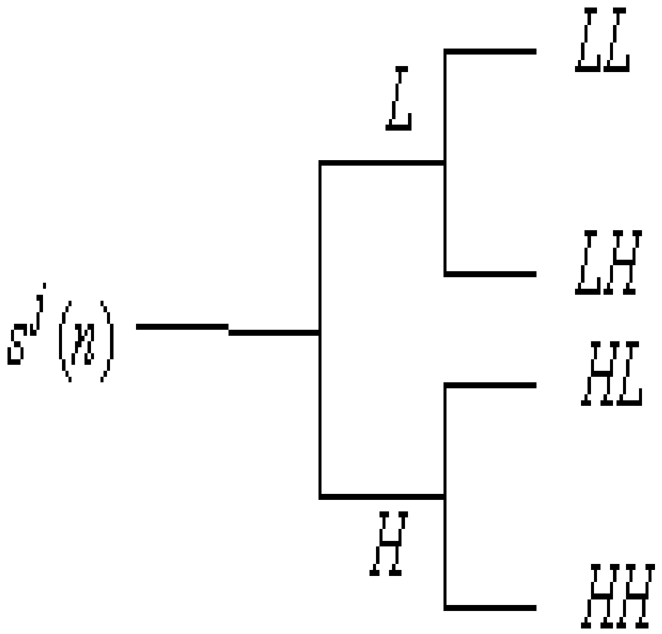

Figure 3.2: Block Diagram for 2D separable wavelet transform. Superscript j indicates level. L and H, respectively indicate lowpass and highpass subands.

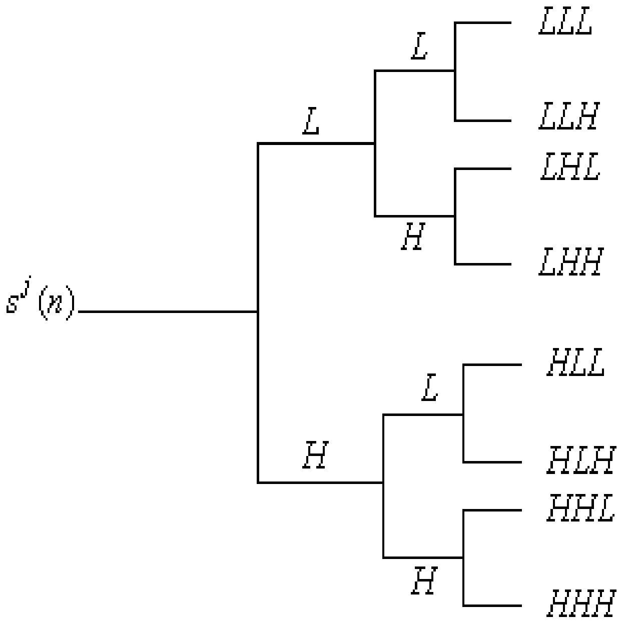

Figure 3.3: Block Diagram for 3D separable wavelet transform. Superscript j indicates level. L and H, respectively indicate lowpass and highpass subands.

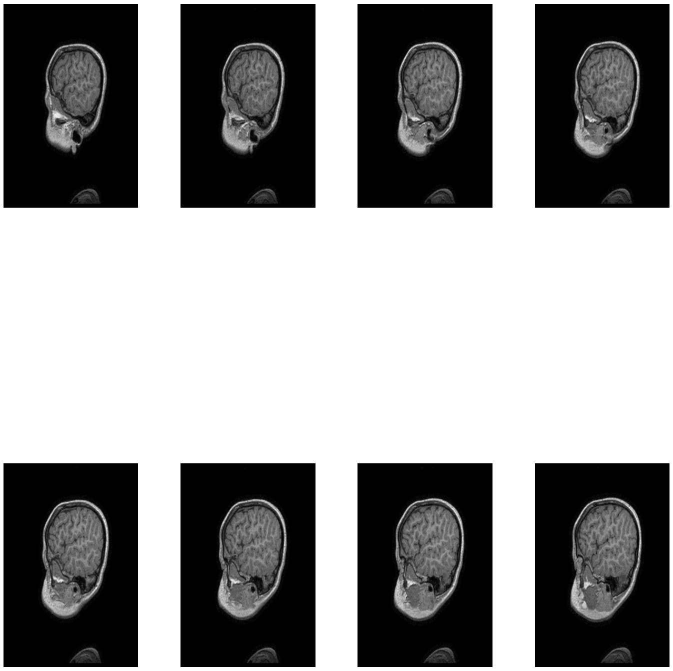

Figure 3.4: Sample 8 consecutive slices of Brain MR Images with 1mm thickness.

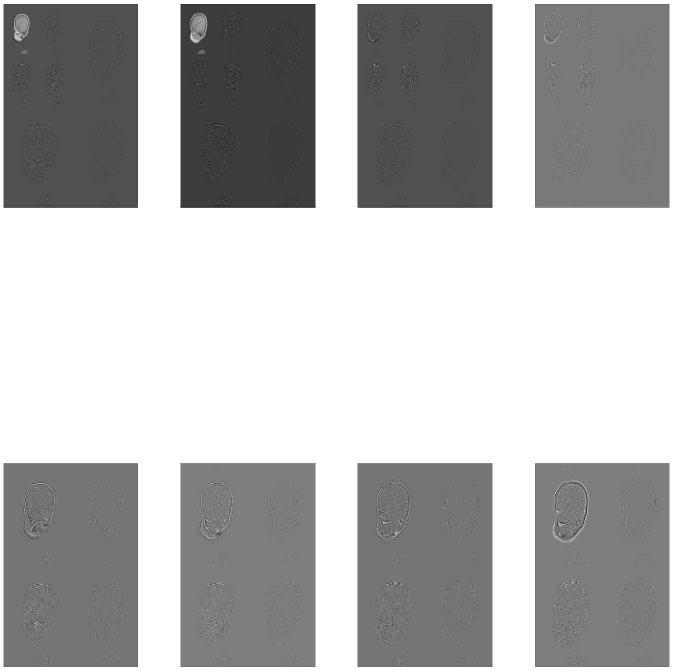

Figure 3.5: 3-D Separable Integer Wavelet Decomposition of the MR Images shown in the Figure 3.4 with 2 levels in spatial and 2 levels in temporal (across slices) domain.

where w_{max} is the absolute maximum of all the wavelet coefficients. A wavelet coefficient is said to significant with respect to a threshold t_n if its absolute magnitude is greater than or equal to a threshold t_n otherwise it is termed as insignificant with respect to the threshold t_n. Let the set D_n contains all the indices of wavelet coefficients which are insignificant with respect to t_n, where $t_n = t_0/2^n$. Let the set R contains indices of all coefficients which are significant with respect to t_n. The embedded bit stream can be generated using the following algorithm:

Choose a predefined scan path and set $n = 1$.

1. Set $n = 1$ and $t_n = t_{n-1}/2$.

2. If a wavelet coefficient w belongs to D_{n-1} (ie insignificant wrt t_{n-1}), encode 1 if $|w| \geq t_n$ and enode its sign (0 if positive or 1 if negative), else encode 0. This is called a *significance pass.* Move this coefficient to the set R. Else if w belongs to R, encode the n^{th} most significant bit (MSB). This step is called *refinement pass.*

3. set n = n+1 and repeat the above steps until $t_n = 0$ or a target bit rate is met.

The significance pass corresponds to encoding MSB of a wavelet coefficient. At the decoder, each one of the above step can be reversed and coefficients can be reconstructed to get an approximate representation of the wavelet coefficients. An exact representation is possible when integer wavelet transform is employed and the bits are encoded and decoded upto $t_n = 0$. This results in lossless compression of the data. Usually the symbols are encoded using entropy coders like run length coding, Huffman coding or Arithmetic coding to exploit the statistical redundancy in the symbols. The bitplane coder is inefficient because it does not exploit correlation among wavelet coefficients. Efficient coding schemes like EZW, SPIHT, LZC and EBCOT exploit inter and intraband correlation of wavelet coefficients. For example EZW and SPIHT exploit interband correlation among the coefficients. They are based on the assumption that if a wavelet coefficient is insignificant wrt to threshold, then all the children along the same spatial orientation can be expected to be insignificant. In such a case all these coefficients will be represented by a special symbol called *zerotree.* If the number of zero trees are more, the compression efficiency will be good. On the other hand, coding schemes like EBCOT exploit intraband correlation. This is based on the observation that significant wavelet coefficients appear in clusters. Instead of a applying dominant pass on individual coefficients, it is applied on non-overlapping blocks of wavelet coefficients. If all the elements in a block are insignificant, wrt to a threshold, entire block is encoded by a single symbol. These algorithms employ either fixed- size blocks or quadtree decomposition (variable size block) for exploiting intraband correlation. The recent algorithms [26], [25] on volumetric image compression exploit intraband correlation by extending 2D wavelet based algorithm EBCOT to 3D images.

Recent studies of [22], [30], [36] suggest that exploitation of intraband correlation is better than that of interband correlation. The entropy coding of the symbols can be further improved by using context-based entropy coding schemes.

3.3.2 Algorithm

We propose a new Region of interest (ROI) based coding scheme for 2D and 3D images. We first give the algorithm for 2D images which can be easily extended to 3D images. We use a fixed-size block to exploit intraband correlation. The region of interest coding can be easily incorporated in intraband schemes. In the previous chapter, we gave a simple algorithm to generate a mask to suppress the background of MR images. We use this mask to code only the regions of interest. In our work, we define ROI as the flesh part. To be able to identify the corresponding regions of interest in the wavelet domain, we also decompose the mask. Figure 3.6 (a) and (b) respectively show a MR image and its corresponding mask and (c) and (d) respectively show a 2 level wavelet decomposition of MR image and 2 level decomposition of mask. The ROI for each subband is clearly delineated by the decomposed mask.

1. Apply n-level 2-D separable integer wavelet transform to the given image and label all background pixels with do not care (DNC) symbols.

2. Tile the wavelet transformed image into $v \times v$ lattices.

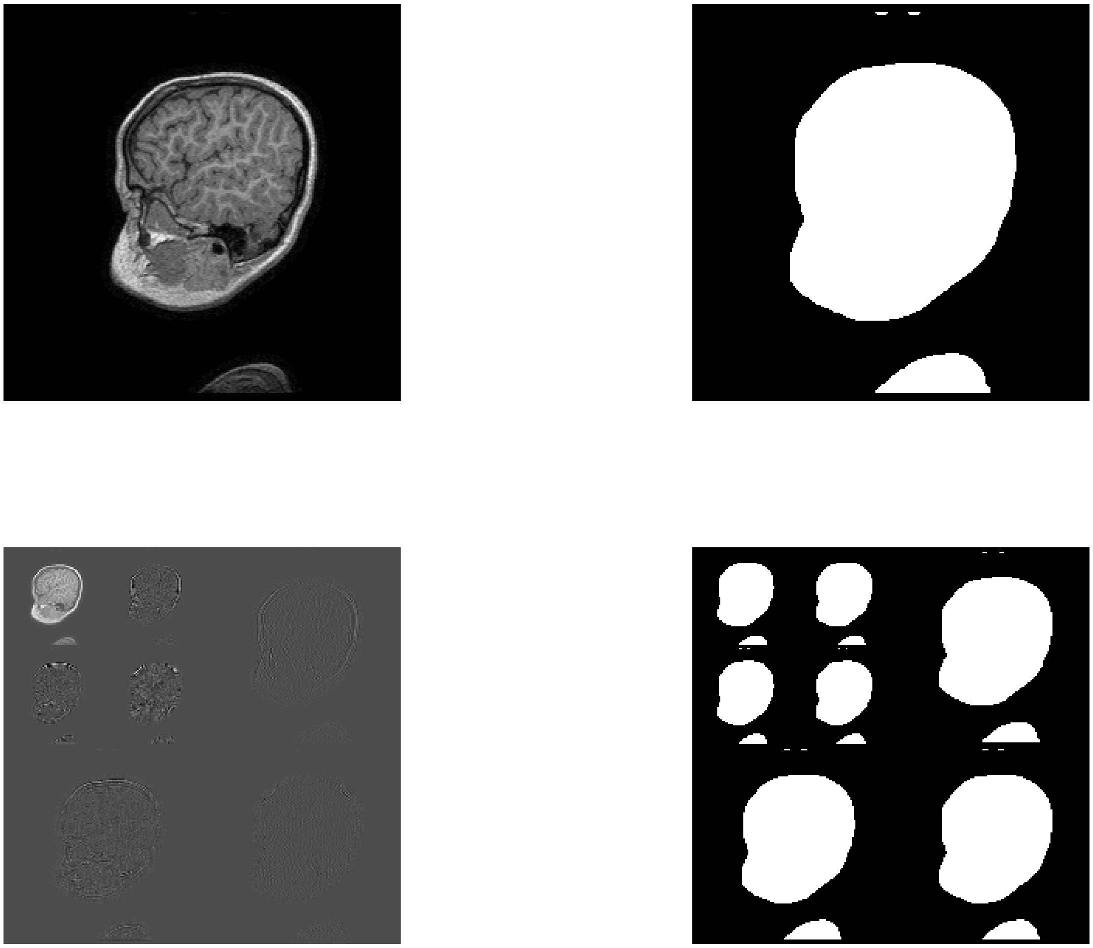

Figure 3.6: (a) MR image (b) its corresponding mask, bright regions show region of interest (c) 2-level wavelet decomposition of (a) and (d) 2-level decomposition of mask in (c) showing regions of interest in each subband.

3. For each lattice k having atleast one non DNC symbol, find the maximum absolute value w_{max} and ignore lattices with all DNC symbols. Let $T_k = |log_2 abs(w_{max})|$ be the threshold of the kth lattice. Store these values in the array th.

4. Set the maximum of all the thresholds as the global threshold, T_g.

5. Scan the wavelet image starting from the lowest frequency band to the highest frequency band in zig zag manner. In each band, the lattices are scanned in a raster order. Lattices with all DNC elements are not scanned.

6. If $T_k < T_g$, the lattice is insignificant with respect to T_g and a '0' is recorded in the list $lstb$. If $T_k >= T_g$, the lattice is significant which decoder needs to be informed . If this lattice is first time significant, a '1' is recorded in the list $lstb$. If the lattice is already significant, no information is sent to the decoder, since this lattice will be significant for the future lower global thresholds.

7. If the lattice is significant, check for the significance of each non-DNC coefficient in raster scan order. If the coefficient is absolute significant, a '1' is appended to the significant list $lis1$ otherwise a '0' is appended. If the coefficient is positive significant, a '0' is appended to the sign list $lis2$ or a '1', if it is negative significant.

8. *Refinement Pass:* The current bit of the significant coefficients (at the previous threshold) is sent to the decoder through the list $lis3$. stop, otherwise go to step 5.

The lists $lis1$, $lis2$, $lis3$ and $lstb$ can be further losslessly compressed by employing arithmetic coding. The mask information is sent as a side information to the decoder. Since the most important coefficients (with higher thresholds T_k) are coded before the least important ones (with lower thresholds), there will be an ordering of wavelet coefficients resulting in progressively transmittable bit stream. The decoder can stop at any step and reconstruct the image that is best at that level. The image reconstructed at $T_g = 0$ will be identical to the original image and hence results in lossless compression. At low thresholds, more blocks will be insignificant and a high value of v results in good compression. But at low thresholds, more coefficients will be significant and if v is high, the cost for sending individual significance of each non-DNC coefficient is high. This impairs compression performance at low thresholds. Hence there is a trade off in the choice of lattice size v. We choose $v = 4$ which is based on our experimentation with different values of v.

3.3.3 Extension to 3-D images

The above scheme can easily be extended to volumetric MR images. The idea is to exploit both intraframe and interframe correlation in these images and hence to achieve higher compression ratios. We extended this scheme to 3-D images by employing separable 3-D integer wavelet transform. A 1-D integer wavelet transform is applied across the slices at each pixel. This results in temporal low frequency and high frequency images. Each of these images are then decomposed by separable 2-D integer wavelet transform as described in section 3. Hence each level of decomposition results in 8 bands (4 temporal lowpass bands and 4 temporal high pass bands) as shown in the figure 3.3. The motivation behind applying 3-D transform is to decorrelate the images across the slice direction in addition to spatial decorrelation so that decorrelated images can be efficiently coded. Region based coding is achieved by decomposing the masks similar to the 2D case. Figure 3.7 shows masks of MR images of 3.4. Figure 3.8 shows 2 level decomposition of masks along both slice and spatial directions. The decomposed masks in various subbands delineate regions of interest. Tile the wavelet images by cuboids of size $v \times v \times vt$ (typically $v = 4$ and $vt = 2$). Apply the algorithm described in the previous section to obtain a progressive bit stream. The coding performance can be further improved by using context-based entropy coding of significance and sign lists.

3.3.4 Context-based Entropy Coding

As mentioned earlier, the lists $lis1$, $lis2$, $lis3$ and $lstb$ are entropy coded using Arithmetic coders. Arithmetic coders are not only efficient than Huffman coders but also have the advantage that source modeling is independent of coding. If the symbols to be coded are assumed to be independent of each other then the lower bound on the number of bits per symbol given by the first order entropy of the source to be coded. Arithmetic coders

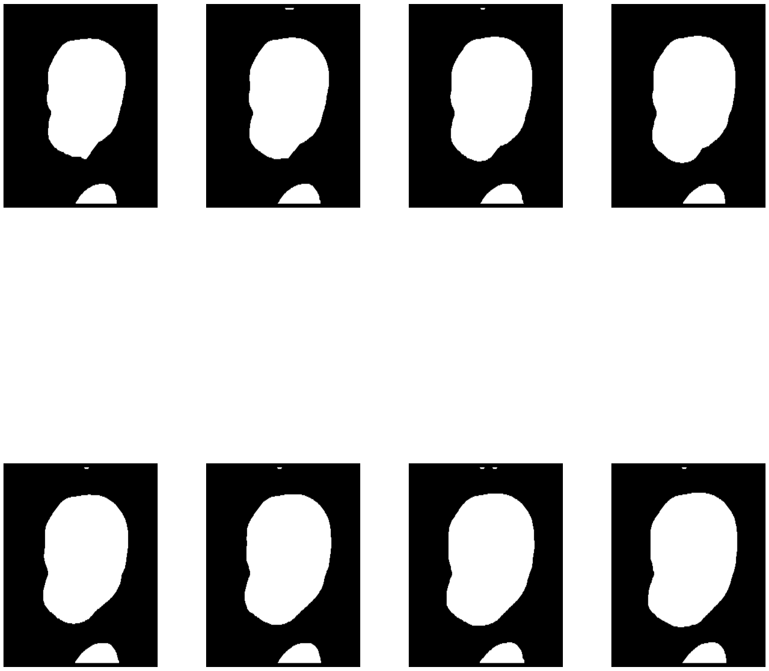

Figure 3.7: Masks showing regions of interest of MR images in 3.4

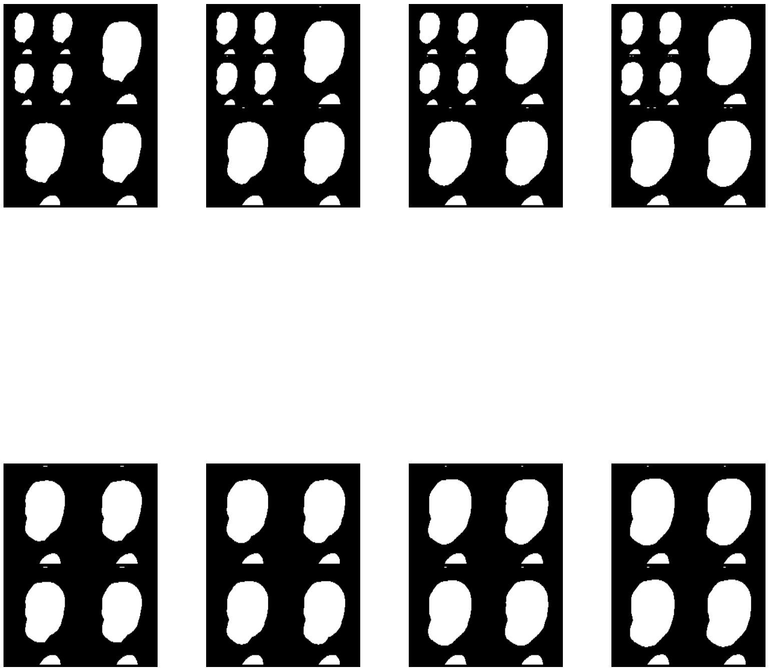

Figure 3.8: 2-level decomposition of masks of 3.7 showing regions of interest in each subband

asymptotically achieve this bound. But in practice, the symbols to be coded are not independent of each other. For example, if a wavelet coefficient w is insignificant, then we can expect the neighbouring coefficients are also insignificant. Also, significance of a coefficient can be predicted using the significance information of its neighbours. Similar observations can be drawn for signs of the coefficients. Using this correlation, one can reduce the lower bound on the achievable bit rate using context-based coding. Let S be the source source symbol to be coded and $H(S)$ denote the first order entropy. Let C be the context or conditioning event and $H(S|C)$ be the conditional entropy of S given the context C. Then one can show that $H(S) \leq H(S|C)$ with equality if and only if S and C are independent. The quantity $I(S;C) = H(S) - H(S|C)$ is called mutual information. The greater the mutual information, the lower is the conditional entropy and better is the coding performance. One can maximize the mutual information by carefully choosing the contexts. Generally contexts are defined based on the already coded discrete symbols . Therefore the number of possible contexts are finite. Theoretically, the mutual information can be maximized by choosing as large number of contexts as possible. But in practice, the number of symbols to be coded are finite because of which some contexts may have few symbols. Since the probabilities of symbols required for entropy coding are estimated on fly, lesser number of symbols result in poor estimates of probabilities. This increases the coding cost and is commonly refered as context dilution. To avoid this, we need to limit the number of contexts without much decrease in mutual information. The process of choosing a fixed number of contexts from the possible contexts is called as context quantization.

A common approach is to project the higher dimensional context onto a linear space (real line) and find appropriate quantization levels on the real line by optimizing certain cost function [29], [27]. Qunatization levels are generally obtained either by using dynamic programming or by employing Lloyd-Max quantizer. These schemes are suboptimal because projection of higher dimensional contextual information onto a linear space results in information loss. In this work, we use the method proposed by [30]. This method operates directly on C for finding the required number of contexts. This method is based on iteratively merging pairs of contexts which result in minimum mutual information reduction. This scheme is based on the result that merging of contexts result in decrease of mutual information. This scheme is used to form 2D and 3D contexts for coding the significance list $lis1$. We employ the contexts for sign list $lis2$ used in EBCOT. We first explain the context merging algorithm.

Let be the random variable Y be the data to be coded and let the random variable X be the context. The mutual information is defined as

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y) log_2 \frac{p(x,y)}{p(x)p(y)} \tag{3.2}$$

where, $p(x,y)$ is the joint probability distribution and $p(x)$ and $p(y)$ are respective marginal distributions of the random variables X and Y. Let P be the initial number of contexts and F be the required number of contexts. The context merging algorithm described in [30] is as follows:

1. Compute the joint probability distribution $p(x,y)$ from the training data.
2. Find the contexts x_i and x_j such that after combining them, the reduction in $I(X;Y)$ is minimum. This is accomplished by pair wise search.
3. Merge x_i and x_j into one context.
4. Set $P = P - 1$. Repeat steps 2 and 3 until $P < F$.

In [30], this algorithm was used for finding contexts for zero coding (ZC) primitive in JPEG2000. For each wavelet coefficient, eight first order neighbours are used to form a context. Each element either assumes 1 if it is significant wrt a threshold or 0 if it is insignificant. Therefore, there will be 256 possible contexts. The above context merging algorithm is used on a training data to reduce the number of contexts. It was found that the coding performance with 3 or 4 contexts was as good as 9 JPEG2000 coding contexts.

We use similar strategy to define contexts for significance map coding for 2D and 3D schemes. In the 2D scheme, we use eight first order neighbours to define a context. Let $w(i,j)$ be a wavelet coefficient whose significance wrt to a threshold T_k needs to be coded.

Let i, j be the coordinates of the coefficient. The first order neighbourhood of $w(i, j)$ denoted by $N(i, j)$, is given by

$$\begin{aligned} N(i,j) = [&w(i-1,j-1), w(i-1,j), w(i-1,j+1), w(i,j-1), \\ &w(i,j+1), w(i+1,j-1), w(i+1,j), w(i+1,j+1)] \end{aligned}$$

Each element if $N(i, j)$ is threshold to 0 or 1 depending on whether its magnitude is less than or equal to T_k. The first four elements of $N(i, j)$ are scanned before $w(i, j)$ and hence they form a causal neighbourhood. The last four elements are scanned after $w(i, j)$ and they form noncausal neighbourhood. To be able to identify the contexts at the decoder, we use threshold the T_k for the causal elements and T_{k+1} for the noncausal neighbourhood. Since the significance wrt to previous threshold T_{k+1} is already known to the decoder in the case of noncausal neighbourhood. We use any four images from the database as training images and form 256 contexts and estimate joint and marginal probabilities of symbols 0 and 1. We then apply the above context merging algorithm to reduce the number of contexts to 8. We also use the same contexts for the other MRI dataset called MR-MRI data set.

In the case of 3D scheme, we define 3D context as combination of 8 first order neighbours and two elements in the previous and future frames at the same spatial location. Let $w(i, j, l)$ be the wavelet coefficient whose needs to be coded. We define 3D neighbourhood $N(i, j, l)$ of the location i, j, l as follows:

$$\begin{aligned} N(i,j,l) = [&w(i-1,j-1,l), w(i-1,j,l), w(i-1,j+1,l), w(i,j-1,l), \\ &w(i,j+1,l), w(i+1,j-1,l), w(i+1,j,l), w(i+1,j+1,l), w(i,j,l-1), w(i,j,l+1)] \end{aligned}$$

Again $w(i, j, l-1)$ belongs to the causal neighbourhood and $w(i, j, l+1)$ belongs to the noncausal neighbourhood and the thresholds are used accordingly so that decoder will also be able to recover the contexts. Now there will be 1024 possible contexts from which we select 8 contexts using the context merging algorithm. We use any 8 consecutive MR images from the data set as training images to built the required contexts. These contexts are used for the two data sets. For coding of sign information $list2$, we use the sign contexts defined in EBCOT for both 2D and 3D schemes. We do not use any contexts for refinement list $list3$ as the statistical redundancy is small for refinement bits.

3.4 Results and Discussion

We apply the 2D and 3D algorithms on the data sets defined in the previous chapter. We first generate the masks required to define the regions of interest using the mask generation algorithm defined in the previous chapter. In this work, we assume the "flesh" part as the region of interest. We do not code the background image which does not contain any useful clinical information. We send the coordinates of the contour of the mask to the decoder as a side information. The coordinates are coded using differential coding and requires about 0.01 *bpp* to send the mask information. At the decoder, the mask is reproduced by filling the region within the contour by $1's$. We compare the performances of coding schemes with and without region of interest. We apply different integer wavelet transforms defined in the earlier section for 2D and 3D schemes and compare their performances. In the 3D scheme, we use the same filter for both spatial and axial (across the slices) decomposition. We built contexts for 2D and 3D schemes using 8 successive images from one data set. We then use these contexts for other images. In the 3D case, we can apply the compression scheme on the whole volume. But this does not allow access of particular images within the volume. Also, large memory size is required to buffer all the images in the volume. A simple solution to both of these problems is to apply 3D scheme on a group of images (GOI). We define group sizes of 8 and 16 and compare their performances. Finally, we compare the performances of these schemes with that of the mesh-based schemes defined in the previous chapter and with the MLZC and 3D EZW schemes of [1] on the data set MR-MRI.

3.4.1 Choice of block size ($v \times v$)

As mentioned earlier, a lower value of v results in good performance at lower thresholds but poor performance at higher thresholds. We evaluate the performance of both 2D and 3D schemes for $v = 2, 4, 8$. Figure 3.9 shows the performance of 2D coder for the above values of v. The performance of the scheme is better at lower thresholds and poor at higher thresholds for $v = 2$ but the reverse is true for $v = 8$. Since at higher thresholds more number of coefficients will be insignificant, choosing an higher value for v results in lower bit rates. But at lower thresholds, more coefficients will be significant and the choice of larger block sizes results in higher bit rates. Hence there is a trade off in the performance of the coder for different block sizes. We choose block size of 4×4 to get a consistently good progressive transmission. Figure 3.10 shows the performance of 3D coder for the above values of v with the value of vt fixed to 2.

3.4.2 Context Formation

We need to choose the number of contexts and a training data to formulate these contexts. To evaluate the choice of training data on the coding performance, we first selected, 4 images for the 2D scheme and 8 images for the 3D scheme, from the NIMHANS dataset and found the overall coding performance for the filter $(4, 4)$ with $v = 4$ and $vt = 2$. We next chose training images from MR-MRI dataset and formed the required number of contexts. These contexts are then used for NIMHANS dataset. The overall difference in the coding performance is only about $0.003 bpp$ for both 2D and 3D schemes. This experiment suggests that coding performance will not be affected by the choice of the training dataset. We also need to choose the number of contexts for both 2D and 3D schemes. We evaluated the performance of both coders for 32, 16 and 8 contexts. Ideally coding performance should improve with the increase in number of contexts. However, due to context dilution problem, the performance deteriorates when the number of contexts are very high. Table 3.1 shows the coding performance for both 2D and 3D coders for the above number of contexts. The difference in coding performance is almost the same for these number of contexts. We fix the number of contexts to 8 for both 2D and 3D

Table 3.1: Lossless compression in (*bpp*) of 2D and 3D wavelet schemes for different number of contexts. 2D-WC: 2D scheme with context, 3D-WC: 3D scheme with context.

Number of contexts	2D-WC	3D-WC
32	2.16	2.08
16	2.19	2.1
8	2.18	2.1

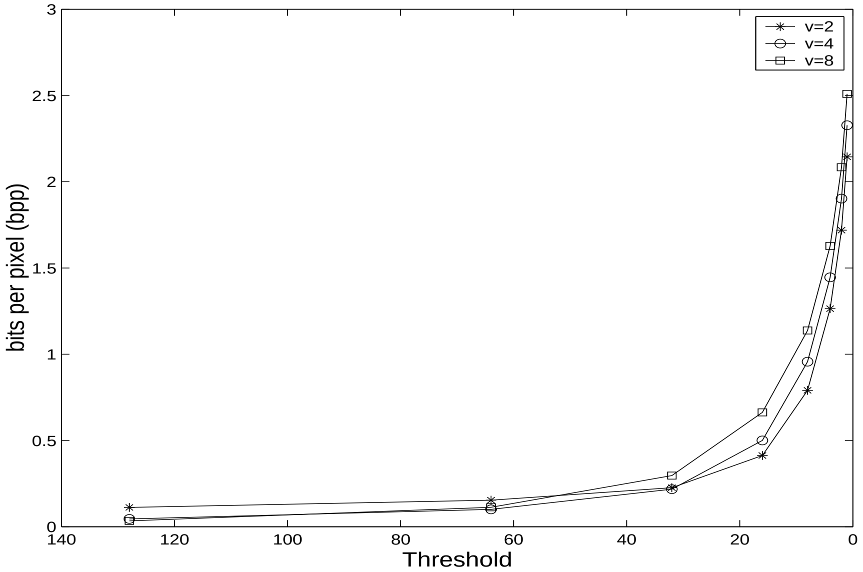

Figure 3.9: Performance of 2D coder for different block sizes $(v \times v)$

schemes.

3.4.3 Performance evaluation for different filters

Table 3.2 gives average lossless compression results in bits per pixel (bpp) for 2D and 3D schemes with and without context based coding on the NIMHANS dataset. We apply 2-level spatial decomposition for 2D scheme and 2-level spatial and axial decomposition for 3D schemes. The group size in the 3D scheme is 8. We fix block sizes as 4×4 and $4 \times 4 \times 2$ for 2D and 3D schemes respectively. All the filters give comparable performances. 3D schemes perform better than the 2D schemes and context-based modeling improves

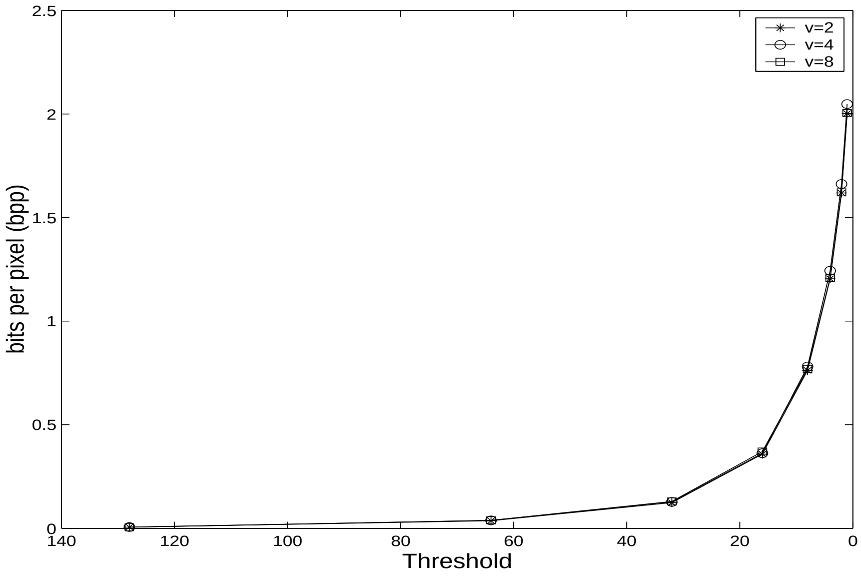

Figure 3.10: Performance of 3D coder for different block sizes ($v \times v \times vt$)

the compression performance by about 0.1 *bpp*. The improvement in compression performance of 3D schemes is about 0.1*bpp* over the 2D schemes.

3.4.4 Rectangular vs Region of Interest Coding

Tables 3.3 and 3.4 respectively give the performance measures in *bpp* on the NIMHANS dataset for 2D and 3D schemes with and without region of interest coding. In the later case, we define entire volume as region of interest and we call the coding scheme as "Rectangular Coding Scheme". The filter used is $(4, 4)$, with 2 levels of spatial and axial decomposition and with the same block sizes as mentioned above. Using the region of interest coding, performance of 2D improves by about 0.2 *bpp* over the rectangular coding. In case of rectangular coding, the incorporation of context based coding improves the coding performance by 0.7 *bpp*. This can be attributed to the efficient coding of the significance map which contains large number of zeros due to the background.

In the 3D case, using the region of interest coding improves the coding performance by 0.2 *bpp* over the rectangular coding. In the rectangular coding, the use of context based coding improves the performance by about 0.3 *bpp*.

Table 3.2: Lossless compression in (*bpp*) of 2D and 3D wavelet schemes for different wavelet filters on NIMHANS dataset. 2D-WOC:2D scheme without context coding,2D-WC: 2D scheme with context, 3D-WOC: 3D scheme without context, 3D-WC: 3D scheme with context.

Filter	2D-WOC	2D-WC	3D-WOC	3D-WC
(2,2)	2.35	2.2	2.18	2.12
(4,4)	2.31	2.18	2.17	2.1
(4,2)	2.33	2.18	2.18	2.1
(2+2,2)	2.33	2.18	2.18	2.1
(6,2)	2.33	2.19	2.18	2.11
(9,7)	2.36	2.21	2.21	2.13

Table 3.3: Lossless compression (in *bpp*) of 2D wavelet based schemes with and without region of interest on NIMHANS dataset. 2D-WOC-REC: 2D scheme without ROI and Context coding, 2D-WC-REC: 2D scheme without ROI and with Context coding.

Filter	2D-WOC-REC	2D-WC-REC	2D-WOC	2D-WC
(4,4)	3.12	2.35	2.31	2.18

Table 3.4: Lossless compression (in *bpp*) of 3D wavelet based schemes with and without region of interest on NIMHANS dataset. 3D-WOC-REC: 3D scheme without ROI and Context coding, 3D-WC-REC: 3D scheme without ROI and with Context coding.

Filter	3D-WOC-REC	3D-WC-REC	3D-WOC	3D-WC
(4,4)	2.61	2.3	2.17	2.1

Table 3.5: Comparative performance of 3D schemes for group sizes of 8 and 16 on NIMHANS dataset . (4, 4) filter is used for both the group sizes.

GOI size	3D-WOC	3D-WC
8	2.17	2.1
16	2.16	2.09

3.4.5 Effect of Group Size on the Coding performance

We evaluate the effect of group size on the coding performance of the 3D scheme. We use two group sizes 8 and 16 with 2-levels of spatial and axial decomposition and with the same block sizes as mentioned earlier. Table 3.5 shows the performance comparison in *bpp* for the two group sizes. The increase in performance with a group size of 16 over the group size of 8 is very minimal. Increase in group requires buffering of more slices and hence more memory requirements. Hence we fix the group size to 8.

3.4.6 Overall Comparison

We compare the performances of 2D and 3D schemes with MLZC and 3D EZW of [1] and the schemes A and B of the previous chapter. For the 2D and 3D wavelet schemes, the performances are given for (4, 4) biorthogonal filter, with 2 levels of spatial and axial decomposition and a GOI size of 8. Table 3.6 gives the performance measures in *bpp* and the best performance is highlighted. This scheme also performs better than MLZC and 3D EZW on the MR-MRI data set. Although the MLZC algorithm provides region of interest coding functionality, the performance is given for lossless mode where the entire volume is considered as region of interest. Therefore, comparison should be made between the MLZC and the 3D scheme with out region of interest coding. As shown in the Table 3.6, the performances are similar. Mesh based schemes A and B perform better than the wavelet based schemes for both the datasets. But the wavelet based schemes provide progressive transmission capability which is not provided by the mesh-based schemes with the current implementation.

Table 3.6: Lossless compression Results (in *bpp*) of wavelet based schemes and mesh based schemes. $(4, 4)$ filter is used for wavelet based schemes.

Data Set	3D-WOC	3D-WC-REC	3D-WC	A	B	MLZC	3D EZW
NIMHANS	2.17	2.2	2.1	1.96	**1.89**	-	-
MR-MRI	1.94	2.1	1.83	1.9	**1.81**	2.14	2.27

3.5 Conclusions

In this chapter, we presented a new 2D and 3D wavelet based coding schemes for MR image compression. 2D and 3D separable integer wavelet transforms are respectively used to decorrelate 2D and 3D images. Correlation within the subbands is exploited using fixed size lattices. These schemes provide important functionalities, required for teleradiology, like progressive transmission, region of interest coding and lossless compression. The performance of coding schemes is further improved by incorporating context based coding for the significance and sign maps. 2D and 3D contexts are constructed using a mutual information based context quantization algorithm. The achieved lossless compression performance of the 3D scheme is comparable to the mesh based schemes proposed in the previous chapter and MLZC and 3D EZW algorithms of [1].

Chapter 4

fMRI Time-series Modeling by PPCA

4.1 Introduction

In this chapter, we develop a probability model for fMRI time series using Probabilistic Principal Component Analysis (PPCA). fMRI noise time series contains both uncorrelated and correlated components. The uncorrelated noise is attributed to thermal and scanner noise. Physiological noise sources such as cardiac and respiratory cycles and their aliased versions and subject motion give rise to low frequency trends in fMRI data. For an accurate estimation of Hemodynamic Response Function (HRF), these noise components need to be modeled properly. Using PPCA, the correlated noise components are represented as principal components of the data with the BOLD signal component removed. The HRF is modeled using both parametric and nonparametric models. In the parametric case, a gaussian function is used to model HRF. In the nonparametric case, a smooth FIR filter is used wherein no assumption is made on the shape of the HRF. We study the influence of sampling rate on the estimation of important parameters of HRF, like time to peak and amplitude. We account for the spatial variation of noise characteristics using a bootstrap method for covariance estimation. The number of principal components that need to be used in the model is determined using Akaike information criterion.

We also develop an algorithm to classify the voxels according to their HRF characteristics. This is useful in event related design paradigm where different voxels have different HRFs depending on the type of activation. Also, one can expect that neighbouring voxels have similar responses. This implies that voxels in a neighbourhood belong to the same class. This spatial smoothness in the class labels is modeled by Markov Random Field (MRF). We develop an algorithm to estimate HRF and classify voxels depending on their HRF characteristics. We use gaussian model for HRF and PPCA model for fMRI time series.

These algorithms are tested on both synthetic and real fMRI data. The performance is tested on synthetic noise data at different SNRs. To test the robustness of gaussian model to the variations in HRF shapes, we synthesize HRFs using parametric models like gamma and poisson and test the ability of the gaussian and FIR models to model these functions. We use different performance measures to quantify the performance of the proposed algorithms.

4.1.1 Notation

We use the following notation for this and next chapter. Scalars are represented by lower case letters (for example, x, t, α etc.). Column vectors are represented by bold face letters (for example $\mathbf{h}$, θ etc) and matrices are represented by upper case letters (for example, A, W etc.). Superscript ^{t} represents transpose of a vector or matrix (for example, A^t represents transpose of a matrx A). $N(\mu, \sigma^2)$ represents a gaussian or normal distribution with mean μ and variance σ^2.

4.2 Probabilistic PCA

Principal Component Analysis (PCA) is a widely used tool for data analysis. PCA does not assume any probability model for the data. Tipping and Bishop [67] showed that by assuming a latent variable model for the data vectors, they can be represented in terms of their dominant principal components. This approach is very useful because we not only represent the data in terms of its principal components but also a probability model for the data can be derived. This probability model in turn can be used for tasks like estimation and detection of signals. We use this approach to model the voxel time series of fMR data and model the noise component in fMR time series in terms of its significant principal components.

4.2.1 A Probability Model for PCA

A $d-$dimensional data vector $\mathbf{t}$ can be related to $q-$dimensional $(q < d)$ latent variables $\mathbf{z}$ as:

$$\mathbf{t} = \mu_{\mathbf{t}} + W\mathbf{z} + \mathbf{e} \tag{4.1}$$

where, $\mathbf{e}$ and $\mathbf{z}$ are independent random processes. $\mu_{\mathbf{t}}$ is the mean of the data vectors. By defining a prior probability density function (pdf) to $\mathbf{z}$, the above equation induces a corresponding pdf to $\mathbf{t}$. If we assume

$$\mathbf{z} \sim N(0, I_q)$$
$$\mathbf{e} \sim N(0, \sigma_n^2 I)$$

then $\mathbf{t}$ is also a Gaussian random vector with pdf

$$\mathbf{t} \sim N(\mu_{\mathbf{t}}, WW^t + \sigma_n^2 I) \tag{4.2}$$

where, I_q and I are $q \times q$ and $d \times d$ identity matrices. With the above pdfs for $\mathbf{z}$ and $\mathbf{e}$, we can show that the columns of W are the rotated and scaled principal eigenvectors of the covariance matrix of the data vectors $\{\mathbf{t_n}\}$. With the above model, the observed vector $\mathbf{t}$ is represented as the sum of systematic component $(W\mathbf{z})$ and independent noise component $(\mathbf{e})$. It is shown in [67] that the ML estimate of W and σ_n^2 are given by

$$\begin{aligned} W &= U_q(\Lambda_q - \sigma_n^2 I)^{1/2} R \\ \sigma_n^2 &= \frac{1}{d-q} \sum_{j=q+1}^{d} \lambda_j \end{aligned} \tag{4.3}$$

where the q column vectors in U_q are eigen vectors of the covariance matrix of the data, with the corresponding eigenvalues ($\lambda_j,\ j = 1, 2, ..., q$) in the diagonal matrix Λ_q, and R is an arbitrary rotation matrix. The estimate for σ_n^2 has the interpretation of 'lost' variance in the projection, averaged over the lost dimensions.

4.3 A Probability Model for fMRI time series

The BOLD response of the brain for a given task can be modeled as convolution of the HR function and the input task [42]. The input task $x(t)$ is considered as a binary function of time which has the value '1' during the period of task and '0' during the rest period.

$$y_m(t) = \sum_{k=0}^{K} x(t-k)h_m(k) + w_m(t), \quad t = (K+1)....N \tag{4.4}$$

where, $y_m(t)$ is the observed m^{th} voxel time series, $h_m(t)$ is the system impulse response, $w_m(t)$ is the additive noise component and K is the length of the system response. We model the system response $\mathbf{h_m}$ by parametric and non parametric models independently. In the former case, we model $\mathbf{h_m}$ by a gaussian function and in the latter by a smooth FIR filter. We omit the subscript "m" denoting HRF for each voxel time series for convenience, even though it is computed separately at each voxel. Also, the noise component $\mathbf{w}$ is modeled separately for each voxel time series. The above equation can be written in matrix form as:

$$\mathbf{y} = X\mathbf{h} + \mathbf{w} \tag{4.5}$$

where, X is the convolution matrix of size $(N-K) \times K$, $\mathbf{h}$ is a vector of size $K \times 1$, which represents HRF and $\mathbf{w}$ is a noise vector of size $(N-K) \times 1$. The noise vector $\mathbf{w}$ can be represented using the latent variable model defined in the last section as:

$$\mathbf{w} = W\mathbf{z} + \mathbf{e} \tag{4.6}$$

where $\mathbf{z}$ is a $q-$dimensional $(q < (N-K))$ latent variable and $\mathbf{e}$ is a iid noise component. Assuming Gaussian pdf models for $\mathbf{z}$ and $\mathbf{e}$ as described in the last section, the columns of W are the q principal components of $\mathbf{w}$ corresponding to the first q dominant eigenvalues. The pdf of noise is given as

$$\mathbf{w} \sim N(0, C_y) \tag{4.7}$$

$$where, \quad C_y = WW^t + \sigma^2 I \tag{4.8}$$

Hence the noise vector $\mathbf{w}$ is represented as a combination of systematic component $(Wbfz)$ and an iid component $(\mathbf{e})$. Now, the observed time-series can be represented as a sum of BOLD response, systematic noise component and an iid noise component:

$$\mathbf{y} = X\mathbf{h} + W\mathbf{z} + \mathbf{e} \tag{4.9}$$

Therefore, the probability model for the observed fMRI time-series $\mathbf{y}$ for a given $\mathbf{h}$ is given by:

$$\mathbf{y}|\mathbf{h} \sim N(X\mathbf{h}, C_y) \tag{4.10}$$

The principal components of noise can be computed from the data itself as explained later. A similar model was proposed in [41], which addresses the problem of activation detection for periodic activation (block design paradigm). They assume that the output response signal is also periodic with the same fundamental frequency as the input. They model it as truncated Fourier series with unknown Fourier coefficients. They assume the following model for the voxel time-series:

$$\mathbf{y_i} = A\theta_\mathbf{i} + B\phi_\mathbf{i} + \mathbf{e_i} \tag{4.11}$$

where A is a truncated Fourier series (columns span the signal subspace), $\theta_\mathbf{i}$ is an unknown vector representing the coefficients for the response. B is the unknown matrix whose columns span the noise subspace, $\phi_\mathbf{i}$ is the unknown vector representing the coordinates of the noise space and $\mathbf{e_i}$ is white gaussian noise with unknown variance σ^2. The unknown quantities $\theta_{\mathbf{i}}$, B, $\phi_{\mathbf{i}}$ and σ^2 are determined by Maximum Likelihood Estimation (MLE) by imposing an additional constraint that the signal subspace is orthogonal to the noise subspace ($A^tB = 0$). They found that the columns of B are principal components of the sample correlation matrix of voxel time-series vectors with the signal component removed. There is no restriction on the choice of the matrix A. However, this method cannot be used for event related design paradigm where, the assumption of periodic response is not valid. The proposed PPCA model for fMRI voxel time-series is able to represent the correlated noise components in terms of principal components without the constraints imposed by the method in [41]. In the following two sections, we give both parametric and non parametric methods for estimating HRF.

4.4 Parametric Modeling of Hemodynamic Response Function

Hemodynamic response (HR) refers to the local change in blood oxygenation as an effect of increased neuronal activity. This response can be modeled as an output of a linear filter for a boxcar input $x(t)$. In this section, we model the system response (HRF) by a Gaussian function, where the parameters give a physiological interpretation [37]. The HRF is represented as:

$$h(t) = \eta \exp(-(t.TR - \mu)^2/\sigma^2) \tag{4.12}$$

where, μ is the time lag from the onset of the stimuli to the peak of HR; σ reflects the rise and decay time and η is the amplitude of the response and TR is the sampling period. Figure 4.1 shows a typical HRF along with the significance of the parameters. Let $\theta = [\mu, \sigma, \eta]$ denote the unknown parameters of the HRF.

Now, the estimation of HRF boils down to estimating the unknown parameter θ, for which we use Maximum aposteriori (MAP) estimation. For this, the parameter θ is modeled as a random variable with a known priori pdf. This method allows us to incorporate the prior knowledge of the parameter through the prior pdf. If the prior is properly chosen, one can expect a better estimate of θ. We model μ, σ and η as

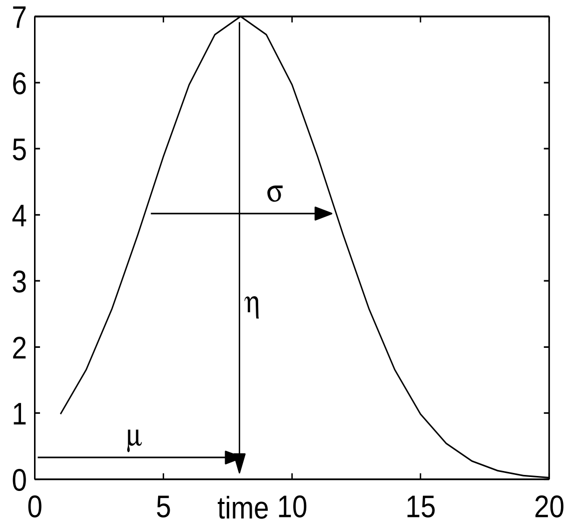

Figure 4.1: An example of Gaussian HR function model. μ represents time to peak, the parameter of interest; σ represents the rise and fall time; and η represents the amplitude of the response which is an indicator of the strength of the response.

independent Gaussian random variables.

$$\begin{aligned} f(\theta) &= f(\mu,\sigma,\eta) = f(\mu)f(\sigma)f(\eta) \\ &= (2\pi)^{-0.5}|V_\theta|^{-0.5}\exp(-0.5(\theta-\mathbf{m}_\theta)^t V_\theta^{-1}(\theta-\mathbf{m}_\theta)) \end{aligned} \tag{4.13}$$

where $\mathbf{m}_\theta$ is the vector of means of μ,σ and η and V_θ is a diagonal matrix whose entries are variances of the above random variables. θ is obtained by maximizing its a posteriori probability given $\mathbf{y}$.

$$\begin{aligned} f(\theta|\mathbf{y}) &\propto f(\mathbf{y}|\theta)\ f(\theta) \\ \theta_{MAP} &= arg\ \ max_\theta\ \ln f(\mathbf{y}|\theta) + \ln\ f(\theta) \end{aligned} \tag{4.14}$$

This MAP estimation is repeated at each voxel time-series. The prior probability parameters $\mathbf{m}_\theta$ and V_θ are chosen using the prior knowledge of the HRF. It is observed that [68] when a stimulus of small duration (unit impulse) is applied, the HRF follows after some delay and attains peak after 2-6 seconds of application of the stimulus. Also, HRF lasts for a duration between 7 to 12 seconds. It is also observed that the amplitude of the response η is around 3-5% of the signal intensity. Accordingly, the priors for μ, σ and η can be chosen as [37]:

$$\begin{aligned} p(\mu) &\sim N(m_\mu, \sigma_\mu) = N(6, 1.732) \\ p(\sigma) &\sim N(m_\sigma, \sigma_\sigma) = N(2, 2.24) \\ p(\eta) &\sim N(m_\eta, \sigma_\eta) = N(4, 2.24) \end{aligned}$$

where, m_μ, m_σ and m_η are means of and σ_μ, σ_σ and σ_η are standard deviations of μ, σ and η respectively. Therefore, the prior probability $p(\theta)$ can be specified as:

$$\begin{aligned} p(\theta) &\sim N(\mathbf{m}_\theta, V_\theta), where \\ \mathbf{m}_\theta &= [m_\mu, m_\sigma, m_\eta]^t = [6, 2, 4]^t \\ V_\theta &= diag(\sigma_\mu^2, \sigma_\sigma^2, \sigma_\eta^2) = diag(3, 5, 5) \end{aligned}$$

4.5 Smooth FIR filter model for Hemodynamic Response function

In this section, we model the HRF by a smooth FIR filter using the approach of [45]. In [45], the noise space is assumed to be spanned by second order polynomial functions. In this work, we evaluate the basis vectors for the noise subspace at each voxel using PPCA.

In modeling HRF by a FIR filter, no specific shape for HRF is assumed. The task is to estimate filter coefficients for each voxel time-series. Since the number of coefficients to be estimated is large, this estimation problem is ill posed. In [45] the problem is regularized by imposing a smoothing constraint on HRF. The HRF is assumed to be a smooth function [68] of time. This smoothness information is translated into the constraint that the norm of second derivative of the HRF $||h(t)''||$, should be small, and then a corresponding pdf is specified as in [45]. The prior pdf of $\mathbf{h}$ takes the following form

$$\begin{aligned} p(\mathbf{h}|\sigma^2,\epsilon) &\propto {\frac{\epsilon^2}{\sigma^2}}^{(K-1)/2} exp(-\frac{\epsilon^2}{2\sigma^2 TR^2}\sum_{k=1}^{K}(h_{k+1}-2h_k+h_{k-1})^2) \qquad (4.15) \\ p(\mathbf{h}|\sigma^2,\epsilon) &\propto {\frac{\epsilon^2}{\sigma^2}}^{(K-1)/2} exp(-\frac{\epsilon^2}{2\sigma^2}\mathbf{h}^t M_p \mathbf{h}) \end{aligned}$$

where, M_p is a concentration matrix given by:

$$\mathbf{M_p} = \frac{1}{(TR)^4}\begin{pmatrix} 5 & -4 & 1 & 0 & \dots & \dots & \dots & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & \dots & \dots & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ \vdots & \dots & & 0 & 1 & -4 & 6 & -4 & 1 \\ & & & & 0 & 1 & -4 & 6 & -4 \\ 0 & & \dots & \dots & 0 & 0 & 1 & -4 & 5 \end{pmatrix} \qquad (4.16)$$

The relative weight of this prior to data is controlled by the hyperparameter ϵ. Also the starting and final values of HRF are assumed to be zero. (i.e., $h(0) = h(K) = 0$). Hence, only $(K-1)$ filter parameters are to be estimated.

Since there is no prior knowledge about ϵ, Jeffrey's prior is assumed [45]

$$p(\epsilon) = (\epsilon)^{-1}$$

Using Bayes rule, the joint posterior pdf for $\mathbf{h}$ and ϵ can be written as:

$$p(\mathbf{h},\epsilon|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{h},\epsilon)p(\mathbf{h}|\epsilon)p(\epsilon) \qquad (4.17)$$

Therefore, the required marginal posterior pdf for $\mathbf{h}$ is given by:

$$p(\mathbf{h}|\mathbf{y}) = \int p(\mathbf{h},\epsilon|\mathbf{y})d\epsilon \qquad (4.18)$$

There is no closed form solution to this marginal pdf and numerical optimization is required to find the MAP estimate of $\mathbf{h}$. To overcome this problem, ϵ is first estimated by MAP and then the a posteriori pdf for $\mathbf{h}$ is approximated as:

$$p(\mathbf{h}|\mathbf{y}) \approx p(\mathbf{h}|\mathbf{y}, \epsilon = \hat{\epsilon}) \tag{4.19}$$

where $\hat{\epsilon}$, the MAP estimate of ϵ is given by:

$$\hat{\epsilon} = arg_{\epsilon} \;\; max \;\; p(\epsilon|\mathbf{y}) \tag{4.20}$$

The marginal posterior probability for ϵ is easy to evaluate and is given by:

$$\begin{aligned} p(\epsilon|\mathbf{y}) &\propto (det(M(\epsilon))^{-1/2} \epsilon^{\frac{(K-1)}{2}} exp(-1/2\mathbf{y}^t C_y^{-1}\mathbf{y} - \hat{\mu_B}^t M(\epsilon)\hat{\mu_B}) \qquad (4.21) \\ where, \quad M(\epsilon) &= X^t C_y^{-1} X + \frac{\epsilon^2}{\sigma^2} M_p \\ and, \quad \hat{\mu_B} &= M(\epsilon)^{-1}(X^t C_y^{-1}\mathbf{y}) \end{aligned}$$

Now, the task is to estimate the HRF $\mathbf{h}$. The fMRI model

$$\mathbf{y} = X\mathbf{h} + W\mathbf{z} + \mathbf{e} \tag{4.22}$$

is a classical bayesian linear model [69]. The posterior probability of $\mathbf{h}$ given $\mathbf{y}$ and an estimate of ϵ can be easily evaluated using the standard results [69]. Since the processes $\mathbf{z}$ and $\mathbf{e}$ are independent with gaussian pdfs and also the prior pdf of $\mathbf{h}$ is gaussian, the posterior pdf of $\mathbf{h}$ will also be gaussian with posterior mean and covariance given by

$$\begin{aligned} E(\mathbf{h}|\mathbf{y}, \hat{\epsilon}) &= (X^t C_y^{-1} X + \frac{\hat{\epsilon^2}}{\sigma^2} M_p)^{-1} X^t C_y^{-1}\mathbf{y} \qquad (4.23) \\ C_{\frac{h}{y}} &= (X^t C_y^{-1} X + \frac{\hat{\epsilon}^2}{\sigma^2} M_p)^{-1} \qquad (4.24) \end{aligned}$$

The conditional expectation of $\mathbf{h}$, $E(\mathbf{h}|\mathbf{y}, \hat{\epsilon})$ is its Minimum Mean Square Estimator (MMSE). The factor $\alpha = \hat{\epsilon^2}/\sigma^2$ determines the relative weight of the prior information.

We get a Maximum Likelihood Estimate for $\alpha = 0$.

4.6 Estimation and Classification Using EM algorithm

In the above two algorithms, each voxel time series is assumed to be independent of other. But we can expect adjacent pixels to have similar HRF. Here, we are not only interested in estimating the HRF but also to group pixels with similar HRFs into same classes. We can expect neighbouring pixels to belong to the same class. This spatial smoothness in class labels of pixels can be modeled using Markov Random Fields (MRF) [70]. Since the class labels are not observable, we can model them as Hidden Markov Random Field (HMRF). We assume that there are K different classes (including no activation) in the given fMRI data. The task is to find the K sets of parameters of the HRF and associate each voxel with one of the appropriate classes. Let Z be the random field of the class labels of the fMRI voxels. The key property of MRF is that the distribution of the random variable associated with a pixel k, given the values of pixels in a neighborhood of k, is independent of the values of the rest of the pixels in the image. This can be written as:

$$f(z_k|z_l, k \neq l) = f(z_k|z_l \in N_k) \tag{4.25}$$

where z_k is the class label of pixel k and N_k is a set of random variables representing the labels for the pixels that are in the neighborhood of k. By Hammersley-Clifford [70] theorem, the distribution over an MRF can be specified in terms of Gibbs distribution

$$f(z) = \frac{1}{Q}\exp(-\sum_c U_c(z)) \tag{4.26}$$

where z is vector of class labels for all the pixels in the image; U_c is the potential function for clique c in the lattice of pixels; Q is the normalization constant. A clique is an ordered set of pixels which are all in the neighborhoods of each other. The sum in the above equation runs over all the cliques as defined by our choice of neighborhood. The potential function gives a potential, or cost, for the particular combination of labels in cliques. We use first order neighborhood where the distance between two pixels is one and each clique contain a pair of pixels. The potential function is defined as

$$\begin{aligned} U(z_k, z_l) &= \beta/2, \quad if \quad z_k = z_l \\ &= -\beta/2, \quad otherwise \end{aligned} \tag{4.27}$$

Hence, the cost for two pixels in a neighborhood to be of different classes increases and is dependent on the parameter β. This will have a smoothing effect on the class label map.

4.6.1 fMR signal model

We use PPCA to model the observed time-series of each voxel. Let there be K different regions (classes) and let $\Theta = [\theta_{\mathbf{1}}, ..., \theta_{\mathbf{K}}]$ denote the parameters of the K HRFs. The observed time-series at mth voxel assuming that it belongs to kth class can be modeled as

$$\mathbf{y_m} = \mathbf{Xh}(\theta_{\mathbf{k}}) + \mathbf{Wx} + \mathbf{e}, \quad \mathbf{m} = \mathbf{1}, ..., \mathbf{M} \tag{4.28}$$

where, M is total number of voxels. Therefore given the class label, the pdf of y_m can be written as

$$p(\mathbf{y_m}|\mathbf{Z_m} = \mathbf{k}, \boldsymbol{\Theta}) \sim \mathbf{N}(\mathbf{Xh}(\theta_{\mathbf{k}}), \mathbf{C}) \tag{4.29}$$

Assuming $\mathbf{y_m}$'s are independent given their class labels (conditional independence), we have

$$p(\mathbf{y_m}|\mathbf{Z}, \boldsymbol{\Theta}) = \prod_{\mathbf{m}=\mathbf{1}}^{\mathbf{M}} \mathbf{p}(\mathbf{y_m}|\mathbf{Z_m} = \mathbf{k}, \boldsymbol{\Theta}) \tag{4.30}$$

The above equation can be maximized wrt $\theta_{\mathbf{k}}$ to estimate θ_k. But the class labels $Z_n = k$ are unknown. Hence the class labels are to be found before the estimation of $\theta_{\mathbf{k}}$. We estimate both class labels and unknown parameters $\boldsymbol{\Theta}$ in an iterative fashion. We first estimate class labels for a given initial Θ^i and using these class labels new Θ^{i+1} is estimated. We repeat this until convergence.

4.6.2 Estimation of Class Labels

The class label map $Z = z$ can be estimated by maximizing the posteriori probability

$$\begin{aligned} f(Z|\mathbf{y}, \mathbf{\Theta^i}) &\propto \mathbf{f(y|Z, \Theta^i)f(Z)} \\ Z = arg\ max_Z\ \ln f(\mathbf{y}|\mathbf{Z}, \mathbf{\Theta^i}) &+ \ln \mathbf{f(Z)} \end{aligned} \quad (4.31)$$

where, Θ^i is an initial value of Θ. The above equation is to be maximized over all possible configurations of Z and hence intractable. It can be approximated using Iterated Conditional Modes (ICM) algorithm [71].

$$\begin{aligned} z_m = arg\ max_{z_m}\ \ln f(\mathbf{y_m}/\mathbf{z_m}, \mathbf{\Theta^i}) + \ln \mathbf{f(z_m/N_m)} \\ f(z_m|N_m) = \frac{1}{Q_m} \exp(-\sum_{j=1}^{4} U(z_m, z_j)) \end{aligned} \quad (4.32)$$

where, N_m is I order neighborhood of the voxel m and Q_m is the normalizing constant.

4.6.3 Parameter Estimation

Using the above estimate of class labels, the parameter Θ^{i+1} can be estimated by maximizing the conditional expectation of the log of posteriori probability $f(\Theta|\mathbf{y}, \mathbf{z})$ $(E\{\ln f(\Theta|\mathbf{y}, \mathbf{z})|\mathbf{y}, \mathbf{\Theta^i}\})$. Using Baye's rule, $f(\Theta|y, z)$ can be simplified as:

$$\begin{aligned} f(\Theta|\mathbf{y}, \mathbf{z}) &\propto f(\mathbf{y}, \mathbf{z}|\mathbf{\Theta})\mathbf{f(\Theta)} \\ &= f(\mathbf{y}|\mathbf{z}, \mathbf{\Theta})\mathbf{f(z)f(\Theta)} \end{aligned}$$

since z is independent of Θ and $f(\Theta)$ is prior probability of Θ $(\Theta \sim N(m_\theta, V_\theta))$. Using the above equation, the required conditional expectation can be written as:

$$\begin{aligned} E\{\ln f(\Theta|\mathbf{y}, \mathbf{z})|\mathbf{y}, \mathbf{\Theta^i}\} &\propto \mathbf{E\{\ln f(y|z, \Theta)|y, \Theta^i\}} \\ &+ E\{\ln f(z)|\mathbf{y}, \mathbf{\Theta^i}\} + \mathbf{E\{\ln f(\Theta)|y, \Theta^i\}} \end{aligned} \quad (4.33)$$

The second term in the above equation is independent of Θ and does not influence the maximization of the required conditional expectation. The third term can be simplified, by taking the conditional expectation wrt $f(l|\mathbf{y}, \mathbf{\Theta^i})$, as follows:

$$\begin{aligned} E\{\ln f(\Theta)|\mathbf{y}, \mathbf{\Theta^i}\} &= \ln f(\Theta) \sum_{l=1}^{K} f(l|\mathbf{y}, \mathbf{\Theta^i}) \\ &= \ln f(\Theta) \end{aligned}$$

Using the above equation, the required conditional expectation can be written as:

$$E\{\ln f(\mathbf{y}, \mathbf{z}, \mathbf{\Theta})|\mathbf{y}, \mathbf{\Theta^i}\} \propto \sum_{\mathbf{l=1}}^{\mathbf{K}} \ln \mathbf{f}(\mathbf{y}|\mathbf{z}, \mathbf{\Theta})\mathbf{f}(\mathbf{l}|\mathbf{y}, \mathbf{\Theta^i}) + \ln \mathbf{f}(\mathbf{\Theta}) \tag{4.34}$$

To maximize the above equation, we need the posteriori probability $f(l|\mathbf{y}, \mathbf{\Theta^i})$ which can be written as

$$f(l|\mathbf{y}, \mathbf{\Theta^i}) = \frac{\mathbf{f}(\mathbf{y}|\mathbf{z} = \mathbf{l}, \mathbf{\Theta^i})\mathbf{f}(\mathbf{l})}{\mathbf{f}(\mathbf{y})}$$

The total probability $f(\mathbf{y})$ of voxels is difficult to evaluate. By MRF property, we know the class label of a given voxel is dependent on its neighborhood. Using this, the above posteriori probability can be approximated as

$$\begin{aligned} f(l|\mathbf{y}, \mathbf{\Theta^i}) &\approx f(l|y_m, \Theta^i, N_m) \\ &= \frac{f(\mathbf{y_m}|\mathbf{z_m} = \mathbf{l}, \mathbf{\Theta^i}, \mathbf{N_m})\mathbf{f}(\mathbf{l}|\mathbf{N_m})}{f(\mathbf{y_m})} \\ f(\mathbf{y_m}) &= \sum_{l=1}^{K} f(\mathbf{y_m}|\mathbf{z_m} = \mathbf{l}, \mathbf{N_m})\mathbf{f}(\mathbf{z_m}|\mathbf{N_m}) \end{aligned} \tag{4.35}$$

The conditional expectation can now be written as:

$$E\{\ln f(\mathbf{y}, \mathbf{z}, \mathbf{\Theta})|\mathbf{y}, \mathbf{\Theta^i}\} \propto \sum_{\mathbf{l}}^{\mathbf{K}} \sum_{\mathbf{m=1}}^{\mathbf{M}} \mathbf{f}(\mathbf{l}|\mathbf{y_m}, \mathbf{\Theta^i}, \mathbf{N_m}) \ln \mathbf{f}(\mathbf{y_m}|\mathbf{z_m} = \mathbf{l}, \mathbf{\Theta^i}) + \ln \mathbf{f}(\mathbf{\Theta}) \tag{4.36}$$

This is maximized wrt Θ. With this new estimate of Θ, the class labels are estimated. This procedure is repeated until convergence. The procedure can be summarized as follows:

1. Initialize Θ_o, $i = 0$
2. Find the activation map using (4.32).
3. Using these class labels find new Θ^{i+1} using (4.36).
4. Go to step 2 until convergence.

4.7 Covariance Matrix and Model Order Estimation

4.7.1 Covariance Matrix Estimation

We need to find noise covariance matrix C_y to estimate HRF using the above methods. For this, we need to remove the BOLD signal component, if present, from the observed voxel time series. One way of estimating C_y is to remove the signal component using least square estimation from the fMRI voxel time-series [48]. But time series contains low frequency trends which impairs the least square performance. Also, we do not know apriori the nature of trend components. To overcome this problem, we employ an iterative procedure to compute the covariance matrix. We first assume that trends are linear and find this linear trend component using the time series data from the rest period (time duration where there is no activation). We remove this estimated linear trend from the original obeserved time series to obtain the detrended time series $\mathbf{y_d}$. We now use least squares method to obtain HRF and then time series without BOLD response as follows:

$$\hat{\mathbf{h_{LS}}} = (X^t X)^{-1} X^t \mathbf{y_d}$$

$$\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\mathbf{h_{LS}}} \tag{4.37}$$

and use sample covariance estimator

$$C = \sum_{0}^{M-1} \tilde{\mathbf{y_p^t}} \tilde{\mathbf{y_p}} / \mathbf{M} \tag{4.38}$$

as in [41], where, M is the number of voxel time-series used in the estimation. We obtain the required covarinace matrix C_y by using eigen value decomposition of C. Using this C_y, we compute HRF by gaussian or FIR filter method. This refined HRF is then used to estimate the BOLD response and is removed from the obeserved time series to obtain time series without BOLD response. This procedure is repeated until the convergence of estimated HRF. Our simulations suggest that this procedure converges in 2 to 3 iterations. This method assumes that the noise characteristics do not change from region to region. However, this is not true [72]. One way to overcome this problem is to estimate the covariance matrix locally using neighboring voxels. We propose a new method for covariance matrix estimation using a bootstrap method in wavelet domain wherein C_y is estimated at each voxel. Wavelet transforms decorrelate a large class of signals [59]. Hence, the detailed coefficients in the wavelet domain can be assumed to be independent and identically distributed (iid) processes. Hence these coefficients can be exchanged (resampled without replacement). In [59], it was shown that resampling the detailed coefficients does not alter the covariance structure of the signal. We can generate multiple processes from a single process by resampling the detailed wavelet coefficients using the following algorithm:

1. **Initialization:** Estimate the trend component using least squares method. Assume linear trend and estimate its parameters using the data in rest period. Subtract this trend component from the observed data. Remove the signal component from the detrended time-series by using a least square estimate of HRF as in (4.37).

2. Take a n-level Wavelet Transform of $\tilde{\mathbf{y}}$.

3. Resample each detailed coefficient without replacement and take inverse wavelet transform to get another realization of the time-series.

4. Repeat the above steps for the required number of realizations.

Figure 4.2 shows the various steps involved in the estimation of the covariance matrix C. In the initialization step, the linear approximation of trend component is estimated using least squares method. The detrended signal is then used to estimate HRF by a least squares method. The estimated BOLD signal obtained by convolution of stimulation and estimated HRF is then subtracted from the original time series. The obtained residue (or noise only) signal is resampled. The covariance matrix C can be calculated using (4.38). We use daubechies-4 mother wavelet (db4) and perform 4-level decomposition. But this method is computationally very expensive. In this work, we use combination of both methods to estimate the covariance matrix. We tile the image into 5×5 blocks and assume that noise characteristics do not change within this block. Every voxel time-series in each such block is sampled for p number of times. The covariance matrix, computed from the resampled realizations of voxel time-series in the block, is used for all the voxels in the block. A number of realizations, M, required for estimating C, equal to two times the length of the time-series to get a reliable estimate of C (Therefore, $p = M/25$). We find C_y by taking the eigen value decomposition of C. C_y from the q eigen vectors is evaluated using (4.8). The number of principal components q to be retained is estimated by Akaike Information Criterion (AIC), which is discussed in the next section. Figure 4.3 shows that the structure of the autocorrelation of a time series does not change with the above resampling algorithm. To improve the estimation of covariance matrix, the above procedure can be iterated by replacing the least squares estimate of HRF by the proposed parametric or nonparametric estimates. Our simulation results suggest that the algorithm converges in 2 to 3 iterations.

4.7.2 Model Order Selection

We use the AIC to find the optimal dimension q of noise subspace. The AIC consists of

$$AIC(q) = -\ln\ p(\mathbf{y}|\theta_{\mathbf{MAP}}) + \mathbf{k_f} + \mathbf{q} \tag{4.39}$$

The first term is the negative log-likelihood function evaluated at $\theta = \theta_{MAP}$, k_f is the number of free unknown parameters (3 for gaussian models and K for non-parametric HRF models). The first term gives the measure of how well the model fits the data and the other two terms measure the complexity of the model. We choose q for which AIC is

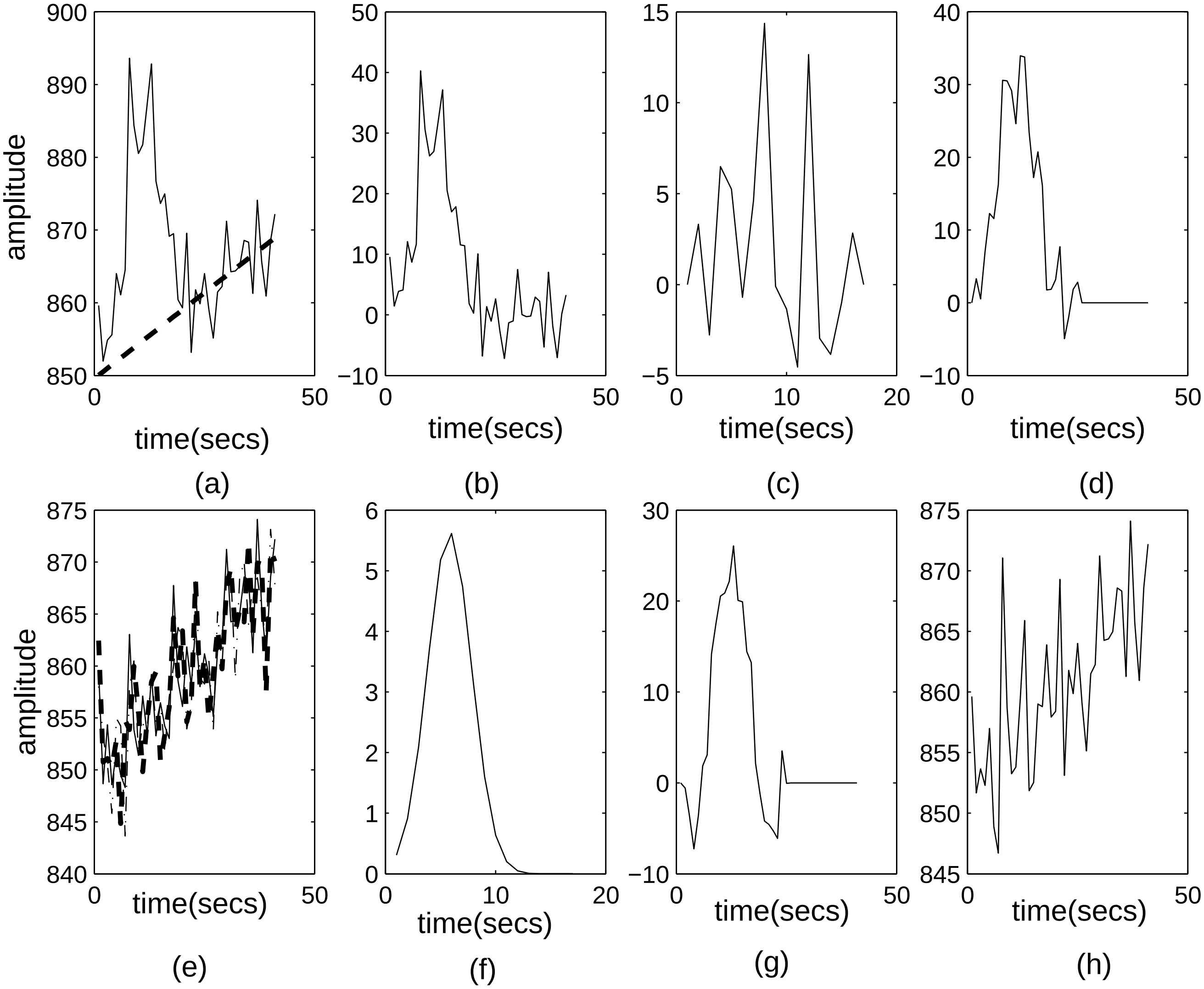

Figure 4.2: Various steps involved in the estimation of covariance matrix. (a) Observed voxel time series. Dotted lines show linear approximation of the trend which is estimated using least squares method. (b) Detrended time series (c) Least square estimation of HRF (d) Estimated BOLD response obtained by convolution of stimulus function and estimated HRF (e) Residue (noise only) time series with estimated BOLD response removed from the observed time series. Dotted signal shows a resampled version of the residue signal (f) Estimated HRF using the parametric model (g) Estimated BOLD response obtained by convolving estimated HRF using parametric model and stimulation function (h) Residue (or noise only) signal obtained by removing the estimated BOLD signal from the observed time series. This residue signal in turn can be resampled for better covariance estimation.

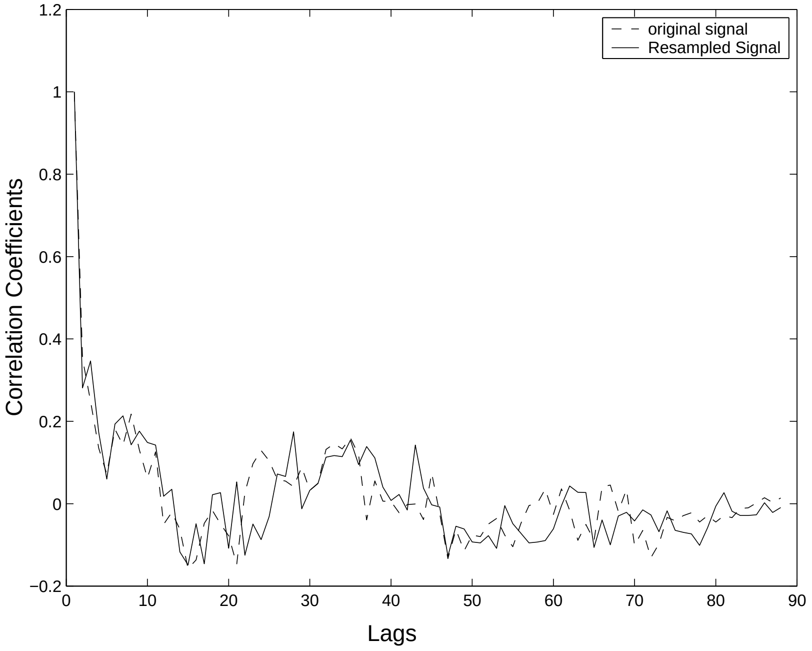

Figure 4.3: Auto Correlation function for original (dotted) and resampled (solid) fMRI voxel time-series. Resampling in wavelet domain does not alter the signal covariance structure.

minimum. Figure 4.4 shows AIC versus q. AIC is minimum for $q = 2$. Figure 4.5 shows the two principal components selected to span the noise subspace. The first component accounts for the base-line drift and the second, the variations in the mean pixel intensity values.

4.8 Test Bed

We test the above algorithms first on a synthetic fMRI noise and then on a real fMRI data obtained when the subject is at rest performing no cogitive or physical task. We call such data as ”Null” data.

4.8.1 Generation of Synthetic data

fMRI time-series exibhits correlation even in the absence of BOLD response. It is observed that [40] fMRI time-series has higher power at low frequencies than at high frequencies

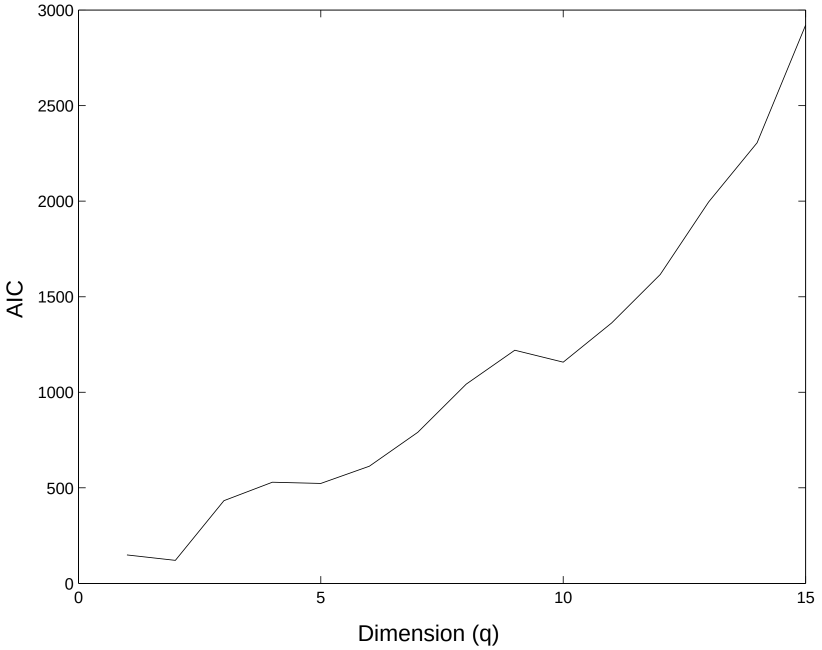

Figure 4.4: Akaike Information Criterion (AIC) verses model order q. Model order is chosen as 2 since AIC is minimum for q = 2.

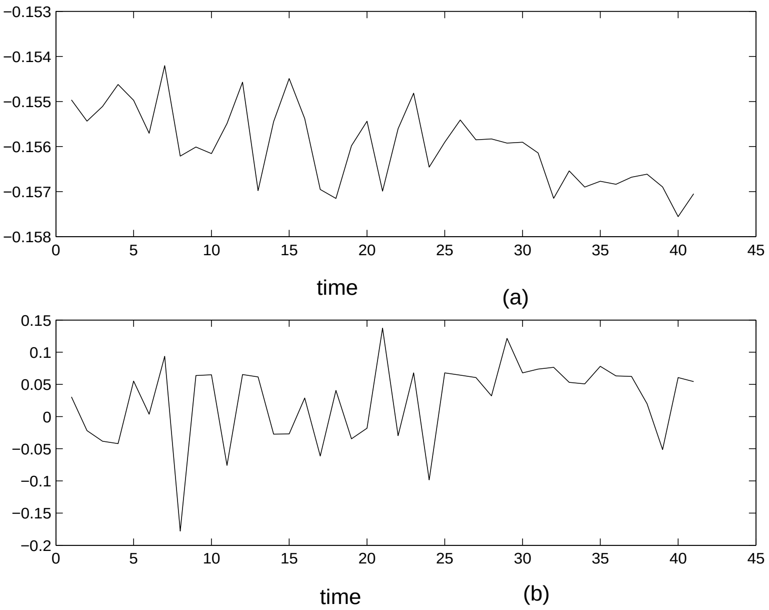

Figure 4.5: Two principal components of Noise subspace chosen according to AIC criterion. The first component (a) accounts for the base-line drift and the second component (b) for variations in the mean pixel intensity values.

and hence has $1/f$-like power spectrum. Most of the real world signals have such spectral properties. Time-series with $1/f$-like power spectrum (also called as $1/f$ process) can be generated using several methods. Wavelet transforms which decorrelate these class of signals can be used to synthesize these signals. We use a wavelet based method suggested in [61]. These signals can be conveniently modeled by fractional Brownian motion (fBm) which is characterized by a parameter called Hurst component (H) (see next chapter for more details on fBm). The variance of detailed coefficients d_j, at each scale j is related to H as

$$var(d_j) = \sigma_b^2 (2^j)^{2H+1} \tag{4.40}$$

This relation is used to generate $1/f$ process. At each scale j, gaussian noise with above variance is generated. The required time-series is generated by taking inverse wavelet transform. We use orthogonal daubechies-4 wavelet with four levels of decomposition. The parameter σ_b^2 is decided by the required signal to noise ratio (SNR). The SNR is calculated as:

$$SNR = 10log_{10} E_s / \sigma_w^2 \tag{4.41}$$

where, E_s is the BOLD signal energy and σ_w^2 is the noise power. We fix the value of H as 0.4. E_s is estimated as $\mathbf{h}'\mathbf{h}$, where $\mathbf{h}$ is HRF, modeled as a gaussian function with typical parameters $\theta = [6, 2, 4]$. Since orthogonal transformation preserves energy, the noise power σ_w^2 is the sum of variances in each scale j.

$$\sigma_w^2 = \sigma^2 \sum_{j=1}^{j=J} (2^j)^{2H+1} \tag{4.42}$$

where, j is the maximum scale allowed. The parameter σ_b^2 can be calculated for a given SNR. We generate synthetic fMR images with size 16 × 16, where each voxel is a time-series of length 64, using the above method. The fMR data is generated at SNRs = 5, 0, -5 and -10 dB. We assume sampling rate (TR) of 1 second and data at $TR = 2$ and $TR = 3$ are generated by subsampling the 1 second data.

4.8.2 Null data

The Null data is provided by NIMHANS, Bangalore. The data is acquired when the subject is at rest condition. The sampling time between two consecutive acquisitions is 1 second ($TR = 1$).

We assume that the subject is at rest between $1 - 10$ seconds, followed by activation between $10 - 20$ seconds and then followed by rest. Thus the experimental paradigm is modeled as a boxcar function with ones between $10 - 20$ seconds and zeros at the other times. The BOLD signal is modeled as a convolution of this boxcar function with HRF as shown in figure 4.6. In the parametric method, we model HRF by a gaussian function. To study the influence of this model on other HRF shapes, we use poisson and gamma functions to simulate HRF in addition to gaussian function and model them all by gaussian function. The convolution of these HRF functions with boxcar function is added to both synthetic and Null fMRI data. In the nonparametric case, we test whether the FIR filter model for HRF is able to model parametric HRF models like gaussian, poisson and gamma functions. We also study the influence of sampling rate on HRF estimation. The performances of these two methods are characterised by the performance indices defined in the next section.

4.9 Results and Discussion

4.9.1 Performance Measures

We use the following performance indices to characterize the performance of the proposed methods. We use gaussian model to fit other parametric functions like poisson, gamma.

1. Time to peak t_p is an important parameter of HRF. It characterises the delay in response to a given task. This parameter can be used to find the relative time of activation of various regions. In our case, since gaussian function is used to model all the types of HRFs, the mean μ of the estimated gaussian function represents the

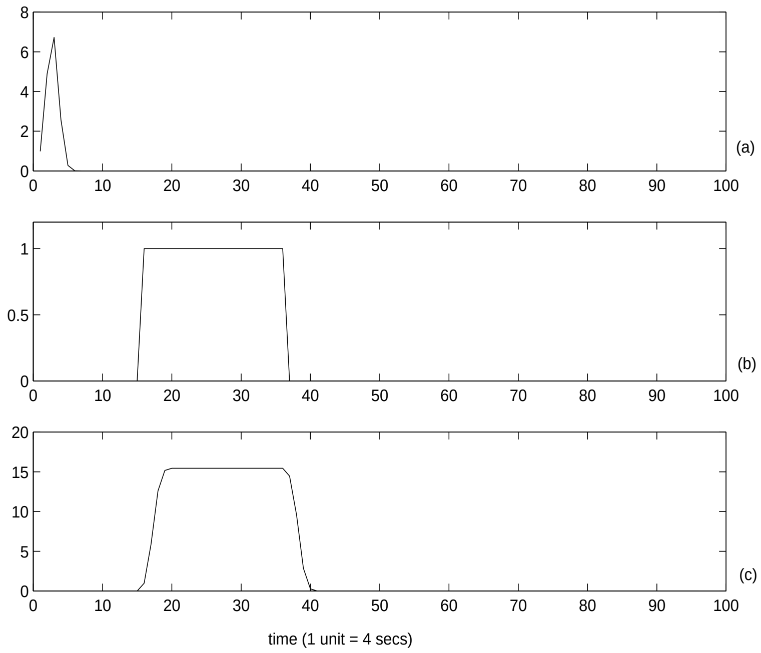

Figure 4.6: Linear model for Blood Oxygen Level Dependent (BOLD) signal (a) HR function (b)Input activation profile or design paradigm (c) BOLD response: convolution of HR and input activation.

time to peak. The percentage error in the estimation of time to peak is defined as:

$$\delta t_p = \frac{|\hat{\mu} - \mu|}{\mu} \times 100 \tag{4.43}$$

where, $\hat{\mu}$ is the estimated time to peak and μ is the actual time to peak. Mean of the gaussian function and parameter of the Poisson model, respectively, characterize the time to peak. In the Gamma model, the ratio of gamma function parameters characterizes the time to peak.

2. The amplitude of the response η to a given stimulus is also an important parameter which characterizes the strength of the response at each voxel. The percentage error in the estimation of this parameter is defined as:

$$\delta\eta = \frac{|\hat{\eta} - \eta|}{\eta} \times 100 \tag{4.44}$$

where, $\hat{\eta}$ is the estimated amplitude and η is the actual amplitude of HRF.

3. The overall error in the estimation of the HRF is characterized by the sample mean square error (MSE) which is defined as:

$$MSE = (\hat{\mathbf{h}} - \mathbf{h})^{\mathbf{t}}(\hat{\mathbf{h}} - \mathbf{h})/\mathbf{K} \tag{4.45}$$

where, $\hat{\mathbf{h}}$ and $\mathbf{h}$ are, respectively, the estimated and actual HRF and K is the length of the response.

4. The sample correlation ρ between the estimated ($\hat{\mathbf{h}}$) and actual HRF ($\mathbf{h}$) measures the match between the estimated and actual HRFs. The range of ρ is from 0 to 1, where 1 corresponds to a perfect match.

In the smooth FIR model, we calculate the time to peak (t_p or group delay) of HRF using

$$t_p = \frac{\sum_{k=0}^{K} kh(k)}{\sum_{k=0}^{K} h(k)} \tag{4.46}$$

The amplitude of HRF is estimated as the peak of the estimated HRF.

4.9.2 Simulation Results for Parametric and Nonparametric Models

Synthetic data

Tables 4.1 and 4.2 summarize the performance of the parametric and FIR filter methods for different SNRs, TRs and HRF models. The parameteric method gives about $14-18\%$ and upto 22.7% error in the estimation of time to peak and amplitude, respectively, at $TR = 1$ for all HRF models and SNRs. The performance is robust for decrease in SNR. The sample correlation is more than 0.8. In the case of FIR model, the estimation error is about $2-4\%$ and upto 14% in the estimation of time to peak and amplitude, respectively, at $TR = 1$ for all HRF models and SNRs down to $-5dB$. The sample correlation is about 0.9. Hence, FIR filter performance is better than the gaussian model

Table 4.1: Performance Comparison of Parametric method at different SNRs and TRs. Poisson,Gamma and Gaussian simulated Hemodynamic Response Functions are modeled by a Gaussian Function. poi: Poisson, gam: Gamma, gau: Gaussian, δt_p: percent error in time to peak, $\delta\eta$: percent error in amplitude, MSE: Mean square error in HRF estimation and ρ: sample correlation between actual and estimated HRFs.

SNR	TR	δt_p			$\delta\eta$			MSE			ρ		
(dB)	(sec)	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
5	1	15	16.5	14	7.6	7.2	10	1.45	1.15	0.7	0.85	0.83	0.87
5	2	24	25	23	12	14.2	13.2	2.0	2.16	2.0	0.64	0.6	0.62
5	3	31	28	30	23	21	22	3.0	2.6	2.81	0.46	0.52	0.44
0	1	16	17	14.5	14	13.6	17.3	1.5	1.3	0.86	0.83	0.82	0.85
0	2	24	25	22.5	15	14	15.35	2.1	2.13	2.1	0.63	0.6	0.62
0	3	29	28	28	20	20.4	21.67	2.4	2.7	2.7	0.5	0 .5	0.5
-5	1	16	17	14.7	20	22.7	21	1.71	1.44	0.97	0.82	0.81	0.84
-5	2	20	21	17	22	21	22	1.73	1.78	1.9	0.72	0.66	0.67
-5	3	23	22	22	31	31	32	2.2	2.1	2.2	0.42	0.43	0.41
-10	1	16	18	15	21	17.4	19	1.68	1.56	1.05	0.82	0.8	0.82
-10	2	18	19.5	15	21	21.3	25.64	2.56	2.74	2.32	0.63	0.61	0.64
-10	3	28	27	27.1	30	31	31	3.6	2.85	3.05	0.43	0.45	0.43

for $TR = 1$. However, at $TR = 2$, the performance of FIR filter model is poorer than the parameteric method. Also, performance decreases with the decrease in SNR. At $TR = 3$, the estimation of HRF becomes very unreliable. Hence, we can conclude that the performanace of FIR model is better for $TR \leq 1$ second (high sampling rate) and for $TR = 1 - 2$ seconds, parametric models are reliable.

Null data

Now, we apply both parametric and nonparametric methods on the Null fMRI data. Figure 4.7 shows the performance of Gaussian model to model Poisson, Gamma and Gaussian parametric functions at $TR = 1$ seconds. The figure shows that the assumed Gaussian model is able to recover the parametric models. Figure 4.8 shows that the FIR filter model is also able to recover the standard parameter models.

Table 4.2: Performance Comparison of nonparametric method at different SNRs and TRs. Poisson,Gamma and Gaussian simulated Hemodynamic Response Functions are modeled by a Smooth FIR filter. poi: Poisson, gam: Gamma, gau: Gaussian.

SNR	TR	δt_p			$\delta\eta$			MSE			ρ		
(dB)	(sec)	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
5	1	2.4	2.17	2.26	6.4	10	14	0.3	0.31	0.4	0.97	0.95	0.93
5	2	25	26.1	28.4	21	27	33	2.8	2.9	2.97	0.52	0.5	0.56
5	3	27	25	28	23	27	35	3.2	4.1	3.9	0.44	0.42	0.45
0	1	3.52	3.12	3.12	7.0	5.65	13	0.42	0.52	0.72	0.95	0.93	0.9
0	2	22	24	26	21	25	32	3.1	3.2	3	0.56	0.51	0.54
0	3	29	30	28	20	24	32	4.32	4.4	4.28	0.44	0.4	0.41
-5	1	4.11	3.2	2	7.1	12	12	0.68	0.68	0.78	0.9	0.9	0.84
-5	2	42	60	49	24	21 19	4.13	4	3.8	0.46		0.45	0.47
-5	3	37	58	55	24	27	25	5.3	5.4	5.2	0.3	0.33	0.34
-10	1	4	4.2	5.3	11.2	7.2	7.1	2.4	2.25	2.6	0.77	0.74	0.7
-10	2	-	-	-	-	-	-	-	-	-	-	-	-
-10	2	-	-	-	-	-	-	-	-	-	-	-	-

Figures 4.9 and 4.10, respectively, show the performance of parametric and FIR modeling methods at TR = 1, 2 and 3 seconds. At TR = 1 sec, the error in the estimation accuracy of time to peak and amplitude is about 15% and 12%, respectively, for all the three simulated HRFs for the parametric method. FIR filter method is able to estimate these parameters with accuracy 5% and 8%, respectively. But at higher TRs, the ability of FIR filter to model HRFs goes down as compared to parametric model which is also evident from the simulated data. The sample correlation ρ between the estimated HRF and actual HRF is about 0.82 for parametric case and is about 0.9 for nonparametric model. Hence at low TRs, both parametric and FIR filter models are able to recover HRFs with good accuracy. At TR = 2 sec, the error in the estimation accuracy of time to peak and amplitude is about 19% and 15%, respectively, for all the three simulated HRFs for the parametric method, whereas FIR filter method is able to estimate these parameters with accuracy of 25% and 17%, respectively. The sample correlation for parametric case is about 0.62 and for FIR case, it is only 0.5. Hence parametric model performs better at higher TRs. In FIR filter model, since the filter length (K) becomes very small

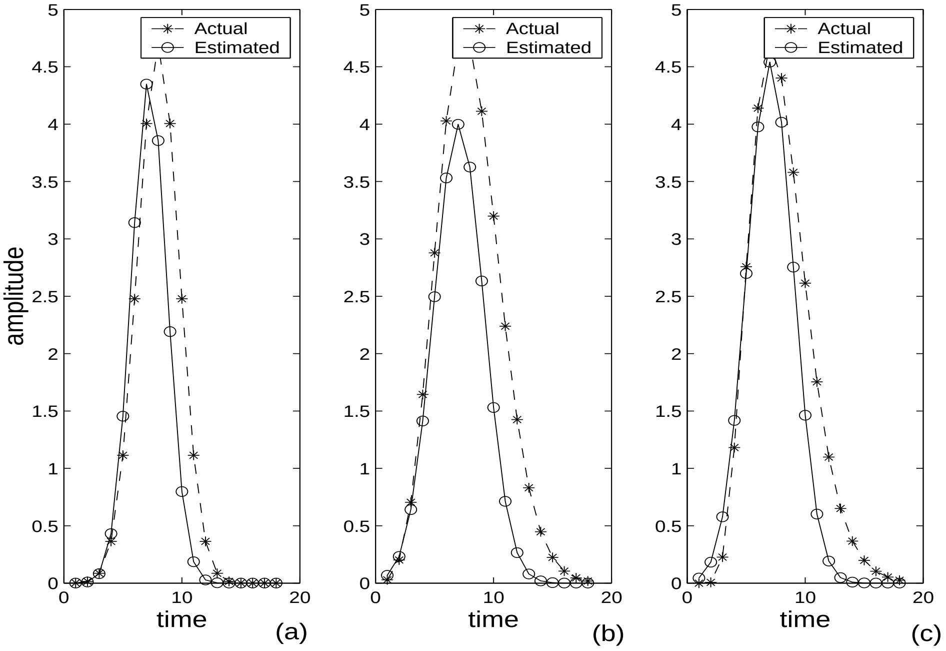

Figure 4.7: Ability of Gaussian HRF model to estimate (a) Gaussian (b) poisson and (c) Gamma Parametric models at $TR = 1$ second.

at higher TRs (K = 7 for TR = 2 and K = 5 for TR = 5), it is difficult to recover the shape of the HRF. Hence, one can conclude that for a reliable estimate of HRF function, low sampling intervals (1 sec and less) are required. At low sampling intervals (low TRs), the FIR filter model will be able to model the HRF without any shape assumption. At moderate sampling rates (TR = 1 to 2 secs), parametric modeling performs better than the nonparametric models.

4.9.3 Simulation Results for Estimation and Classification Algorithm

We apply the estimation and classification algorithm given in section 4.6 on the Null fMRI data. A patch of 16 × 16 is randomly chosen from the fMRI time-series slices. Two gaussian HRF functions with parameters $\theta_1 = [5.5, 2.2, 4.5]$ and $\theta_2 = [7.5, 2.5, 5.5]$ are added to the locations shown in figure 4.11. These two HRFs simulate typical HRF functions from event-related design paradigm. The task is now to estimate and classify

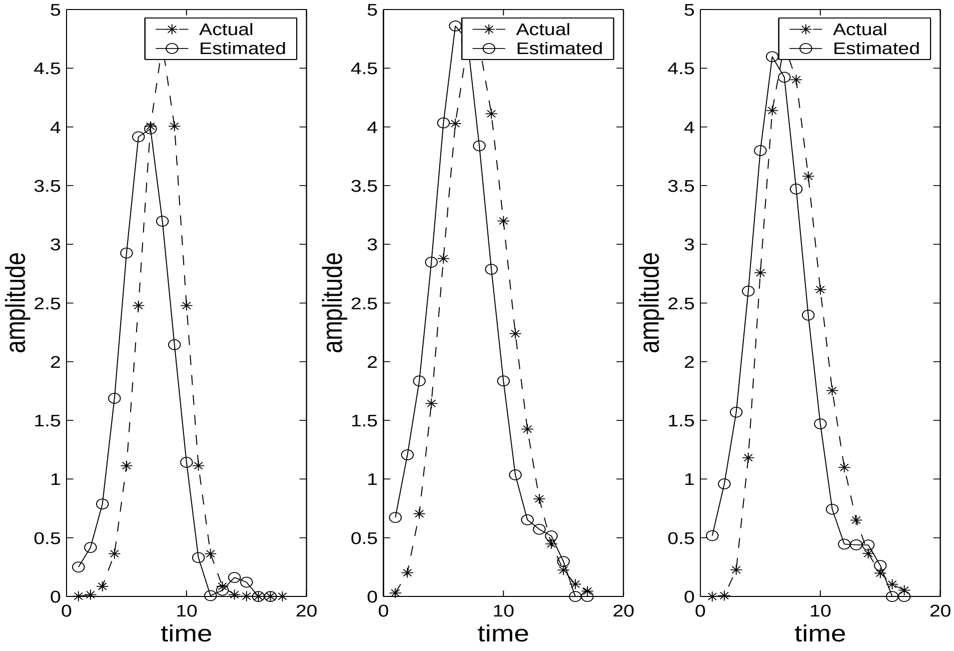

Figure 4.8: Ability of FIR filter to estimate (a) Gaussian (b) poisson and (c) Gamma Parametric models at $TR = 1$ second. FIR filter is able to recover the three parametric models.

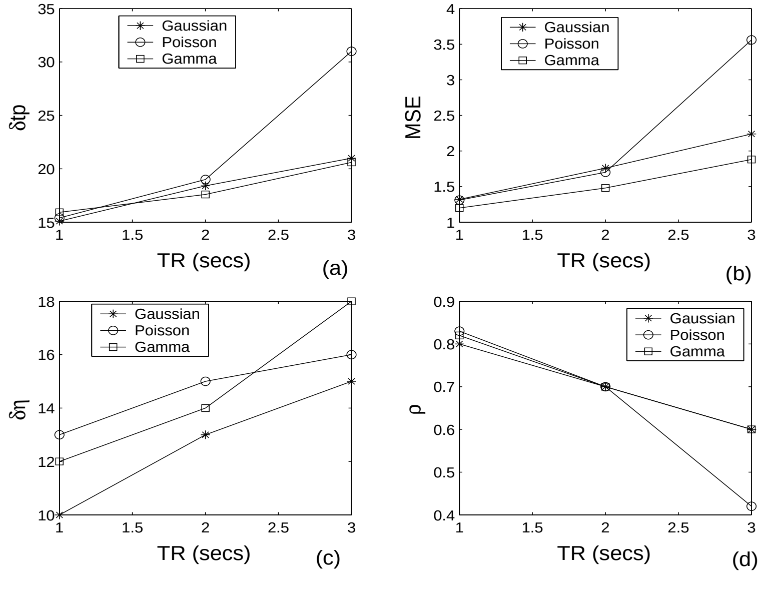

Figure 4.9: Performance measures for parametric modeling of HRF at TR = 1, 2 and 3 secs. (a) Percentage error in the estimation of time to peak ($\delta t_p \times 100$) (b) MSE vs TR (c) Percentage error in the estimation of amplitude of HRF ($\delta\eta \times 100$) vs TR (d) Correlation between estimated and actual HRF (ρ).

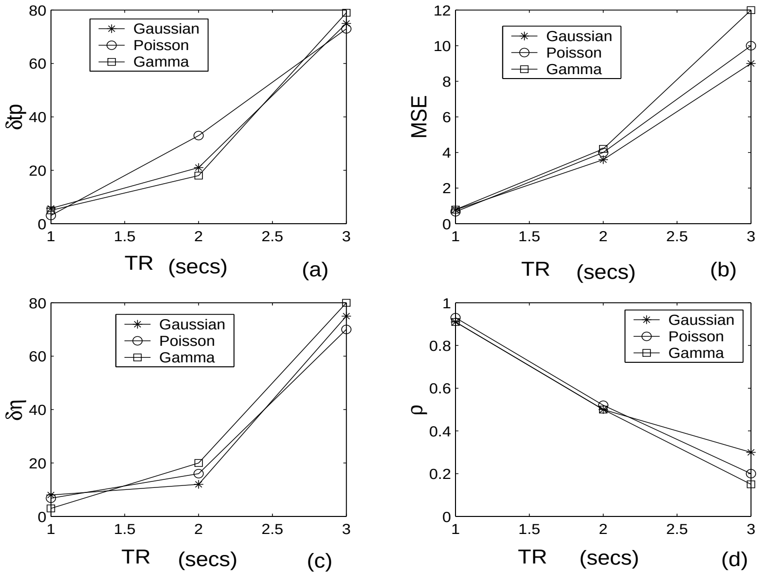

Figure 4.10: Performance measures for nonparametric modeling of HRF at TR = 1, 2 and 3 secs. (a) Percentage error in the estimation of time to peak ($\delta t_p \times 100$) (b) MSE vs TR (c) Percentage error in the estimation of amplitude of HRF ($\delta\eta \times 100$) vs TR (d) Correlation between estimated and actual HRF (ρ).

the given fMRI time series data. This data has three classes, the third class being the non activated voxels shown in figure 4.11 as dark voxels. In addition to simulating gaussian HRFs, we also add poisson and gamma functions and test the ability of gaussian model to model these functions. The parameters chosen for the poisson functions are $\theta_1 = [5.5, 4.5]$ and $\theta_2 = [7.5, 5.5]$, where the first parameter is time to peak and the second parameter is the amplitude of the response function. Gamma functions are simulated with parameters $\theta_1 = [5.5, 2.2, 4.5]$ and $\theta_2 = [7.5, 2.5, 5.5]$. In all the three cases, time to peak and amplitudes of HRFs are the same. We also study the influence of sampling rate (TR) on the estimation and classification accuracy of the alogorithm. Figure 4.12 shows the simulated gaussian, poisson and gamma HRFs at the locations shown in figure 4.11.

Figures 4.13 and 4.14, respectively, show classification and estimation of HRFs for gaussian case at $TR = 1, 2, 3$ secs. First, second and third rows of figure 4.13, respectively, show the convergence of the algorithm at $TR = 1, 2, 3$ secs. First, second and third rows of figure 4.14, respectively, show the estimated and actual HRFs ($h1, h2$) at $TR = 1, 2, 3$ secs. Left and right panels of this figure show estimation of $h1$ and $h2$, respectively.

From these figures, we can infer that classification and estimation performance is good at $TR = 1$ second but degrades for $TR = 2, 3$ seconds. With increase in TR, the number of samples available for the estimation of HRF is small which impairs the estimation of the parameters of HRF which in turn affects the classification performnace. This is because parameters are the important features which govern the classification performance. Signal to noise ratio also decreases with the increase in TR. These are some of the reasons for poor classification and estimation performance of the algorithm at higher TRs.

Figures 4.15 and 4.16, respectively, show classification and estimation of HRFs for poisson case at $TR = 1, 2, 3$ secs. Figures 4.17 and 4.18, respectively, show classification and estimation of HRFs for gamma case at $TR = 1, 2, 3$ secs. At $TR = 1$, gaussian HRF model is able to model and classify both poisson and gamma HRFs. At $TR = 2$ and $TR = 3$, both classification and estimation performance degrades as in the gaussian case. For all the three HRF models, the algorithm converges in 7-8 iterations.

Another important aspect in the algorithm is the choice of number of classes (K). This should be chosen depending on the knowledge of the undelying experiment. If we choose K less than the actual number of classes, we miss underlying fine strutures. Figure 4.19 shows the classification assuming only two classes for the three HRF models at $TR = 1$ second. Top row of this figure shows the classication results for guassian, poisson and gamma HRF models and the bottom row shows the corresponding estimated and actual HRFs. The activated voxels of both classes are classified as one class. Therefore the distinction between voxels with two different HRF functions is lost. Now, we assume number of classes to be four. Figure 4.20 shows the classification for the three HRF models at $TR = 1$ second. The algorithm classifies the data into four classes. Therefore the number of class labels should be chosen properly depending on the underlying experiment. An alternative way of estimating K is by using model order selection methods like AIC. The AIC is sum of two terms:

$$AIC(q) = -l(Y, \hat{\Theta}, Z = z) + q \tag{4.47}$$

where, q is the number of classes. $-l()$ is the negative of the log-likelihood term

Table 4.3: Model order selection for Estimation and Classfication Algorithm. Number of Classes K are selected based on Akaike Information Criterion (AIC) for three HRF models at $TR = 1$ second.

K	HRF	AIC ($\times 10^4$)	HRF	AIC ($\times 10^4$)	HRF	AIC ($\times 10^4$)
2	Gauss	3.6670	Poisson	3.9181	Gamma	3.8349
3	Gauss	3.2278	Poisson	3.2916	Gamma	3.2652
4	Gauss	3.27	Poisson	3.3147	Gamma	3.3086

evaluated at the estimated values of parameters and class labels. Since the voxel time series are independent given their class labels ($Z = z$), l can be written as:

$$l(Y, \hat{\Theta}, Z = z) = \sum_{m=1}^{M} (y_m - Xh(\hat{\theta}_{z_m}))' C^{-1} (y_m - Xh(\hat{\theta}_{z_m})) (4.48)$$

where, M is the total number of voxels and z_m is the class label for the m^{th} voxel. The first term in AIC indicates how well the model fits the data and the second term indicates the model complexity. The optimal model order is the one at which AIC is minimum. Table 4.3 gives the AIC for different number of classes and HRF models. For all the three HRF models, AIC is minimum for $K = 3$ which is the actual number of classes.

We need to also choose the parameter β which governs the spatial smoothness. We study the influence of this parameter on the classification performance. We vary this parameter from 0.5 to 2 in steps of 0.5. Figures 4.21-4.23 show the classification and estimation performance at different βs for gaussian, poisson and gamma HRFs, respectively. Top row of each figure shows the classification performance at $\beta = 0.5, 1, 1.5, 2$ and the second row shows the estimation performance. Both estimation and classification performance does not vary with β. Therefore for all the simulations, we fix $\beta = 1$.

4.10 Conclusions

We presented a new probability model for fMRI time-series using probabilistic principal component analysis. Using this model, the correlated noise (base-line trends etc) in fMRI time-series is represented in terms of principal components of the fMRI data. This probability model is then used to estimate the Hemodynamic Response Function. We modeled HRF by both parametric and non-parametric (smooth FIR filter) models. The number of principal components to be included in the model is determined by Akaike Information Criterion. A new bootstrap method to calculate covariance matrix is also given to take care of spatial noise variation. We also studied the influence of sampling rate (TR) on the accuracy of HRF estimation on both simulated and real fMRI noise data. Smooth FIR filter model is found to be a good method at high sampling rates and parametric model is a better choice at lower sampling rates. We also developed a classification and estimation algorithm using which voxels having similar HRFs are classified together. Classification is performed based on the assumption that neighbouring voxels will have same class labels. This spatial smoothness is imposed by modeling class labels as Markov Random Fields (MRF). Gibbsian distribution is used as prior for class labels. An expectation maximization algorithm is used to determine class labels and HRF functions. PPCA is used to model the fMRI voxel time series. Classification and estimation accuracy is verified for different sampling rates and HRF models.

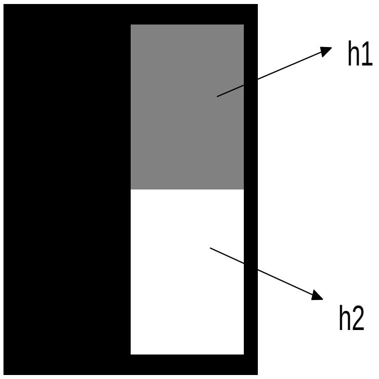

Figure 4.11: (a)Different Regions (Classes) of Activation. Region with dark pixels indicate no activation. Region with gray pixel values indicates activated region with HRF h_1 (with paramter θ_1) and Region with white pixels indicates activated region with HRF h_2 (with paramter θ_2) (b) Actual simulated HRFs h_1 and h_2 with respective parameters θ_1 and θ_2.

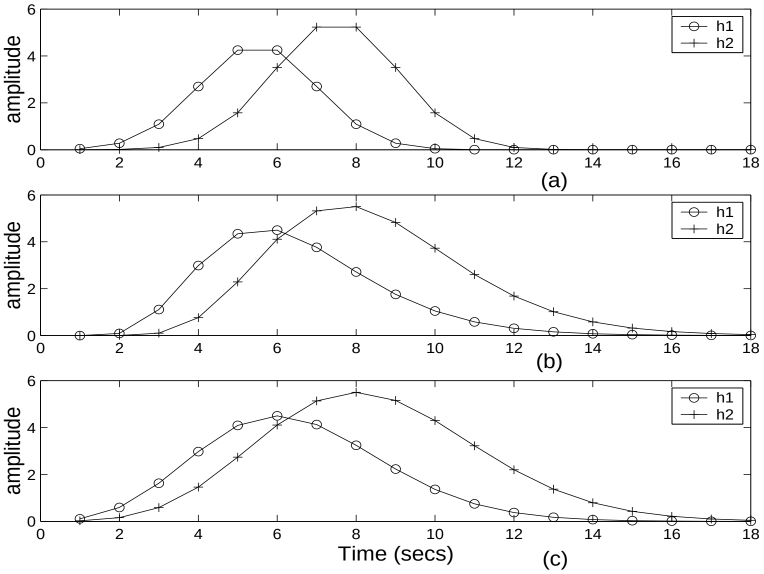

Figure 4.12: Simulated Hemodynamic Response Functions $h1$ and $h2$ (a) Gaussian (b) Poisson (c) Gamma

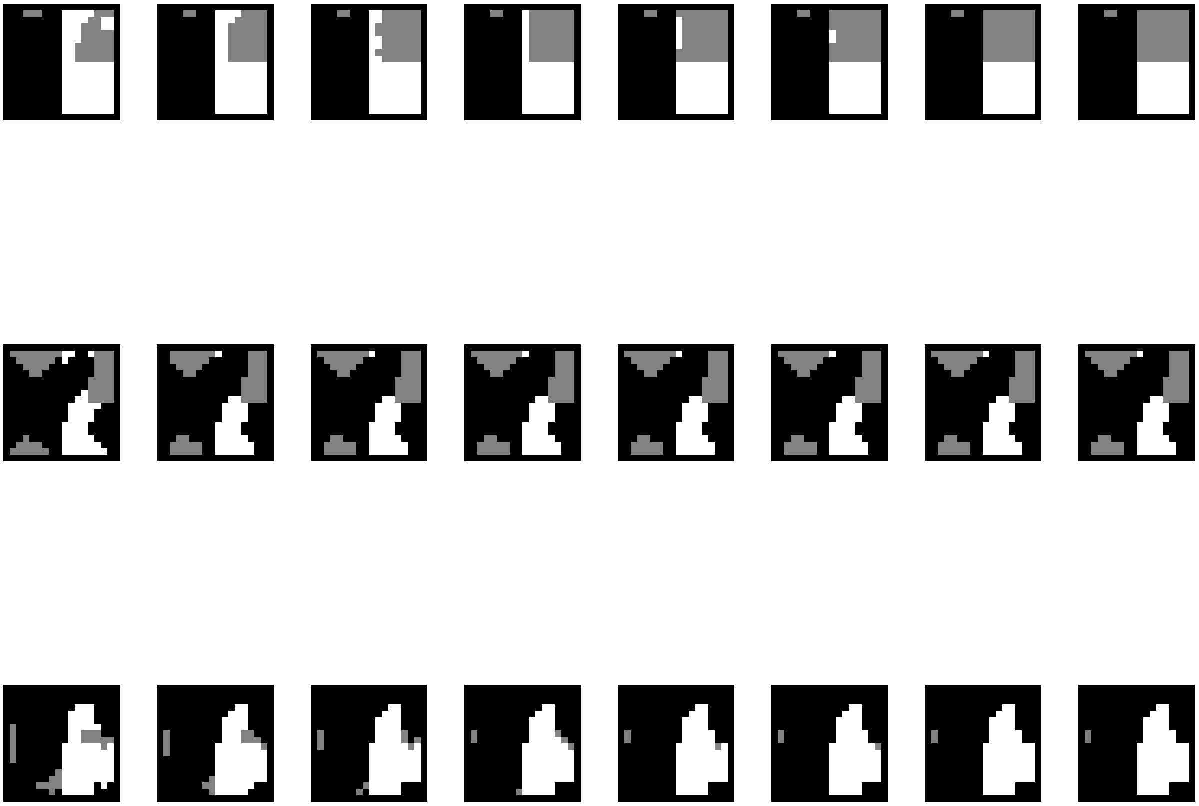

Figure 4.13: Convergence of Classification and Estimation algorithm for the Gaussian simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. The algorithm is able to classify the voxels into three classes for $TR = 1$ second (Row(1)). But for $TRs = 2, 3$ seconds, the algorithm is unable to classify (Row(2) and Row(3)) the voxels.

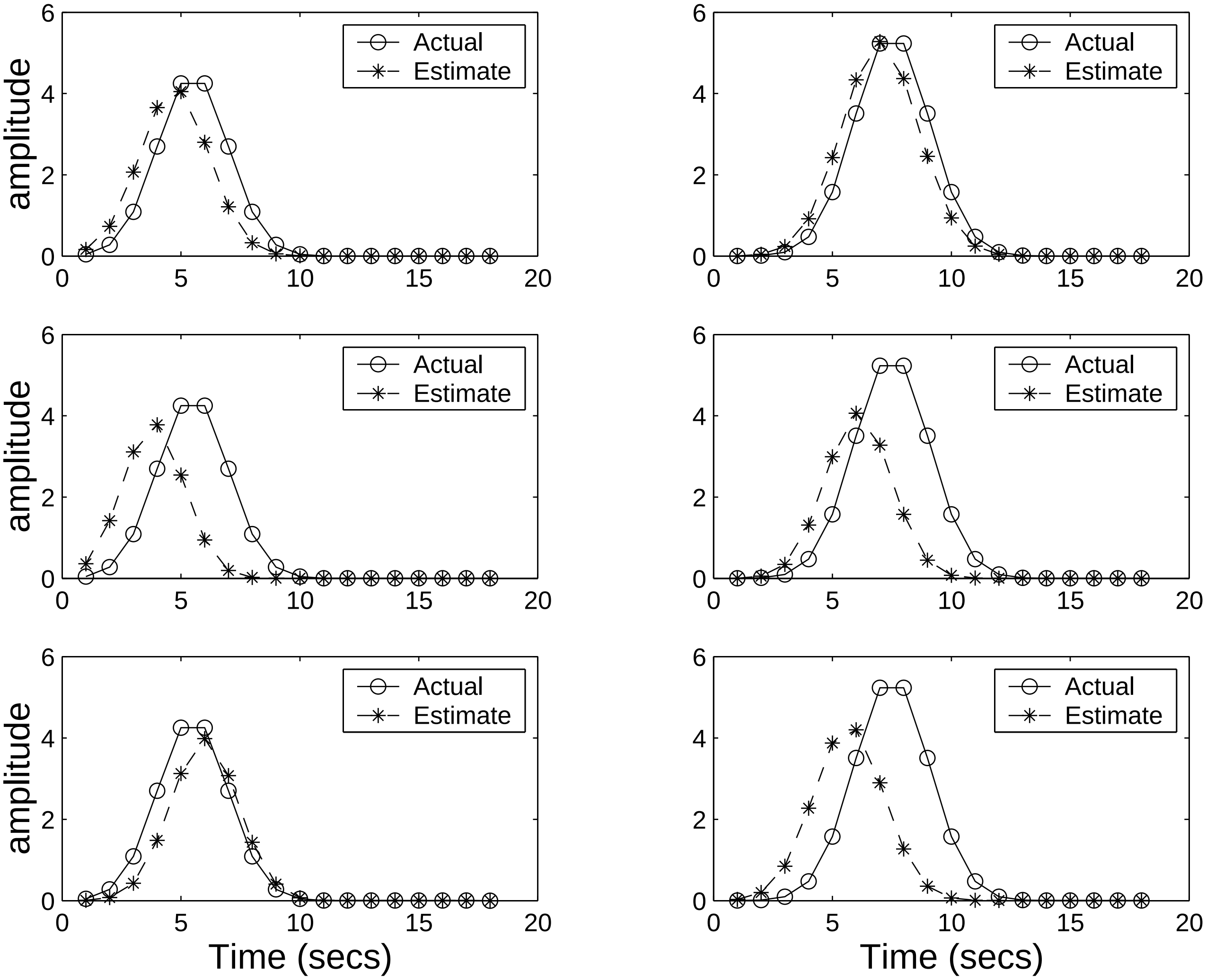

Figure 4.14: Estimation of HRFs of Classification and Estimation algorithm for the Gaussian simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. Left panel shows estimation performance for $h1$ and the right panel for $h2$ at $TR = 1, 2, 3$ seconds.

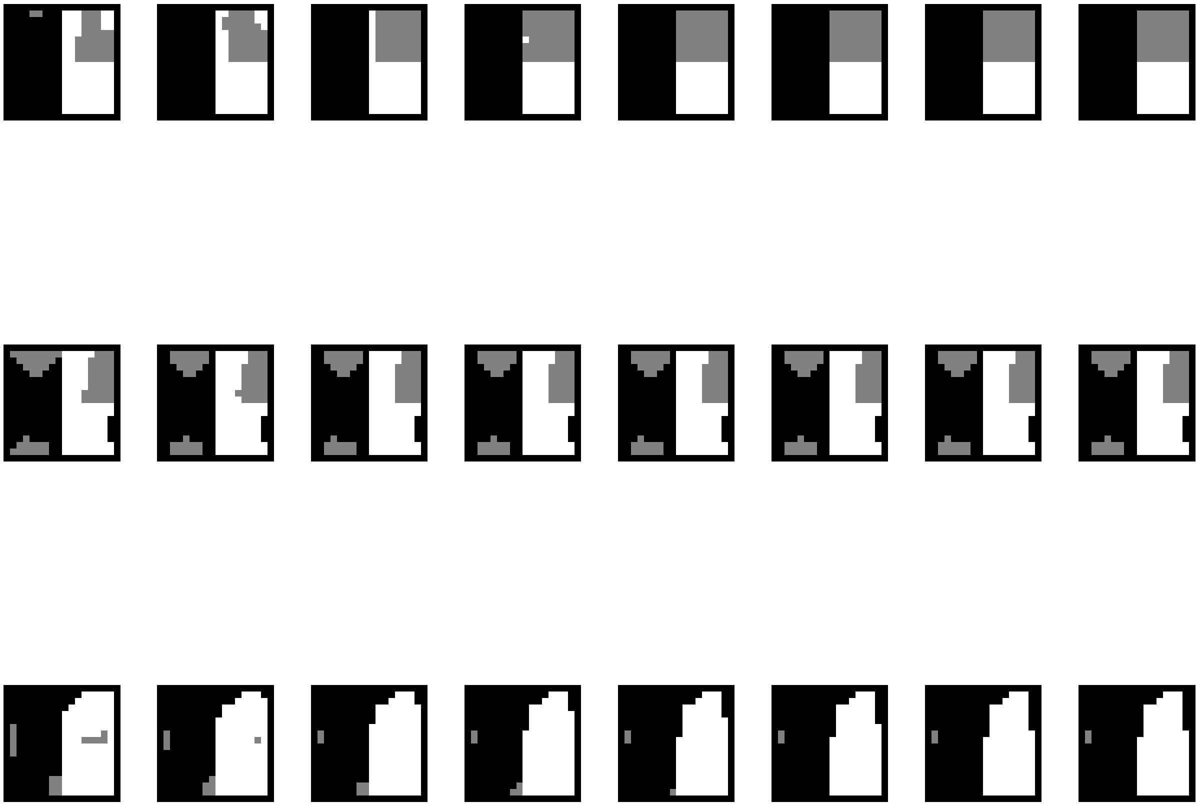

Figure 4.15: Convergence of Classification and Estimation algorithm for the Poisson simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. The algorithm is able to classify the voxels into three classes for $TR = 1$ second (Row(1)). But for $TRs = 2, 3$ seconds, the algorithm is unable to classify (Row(2) and Row(3)) the voxels.

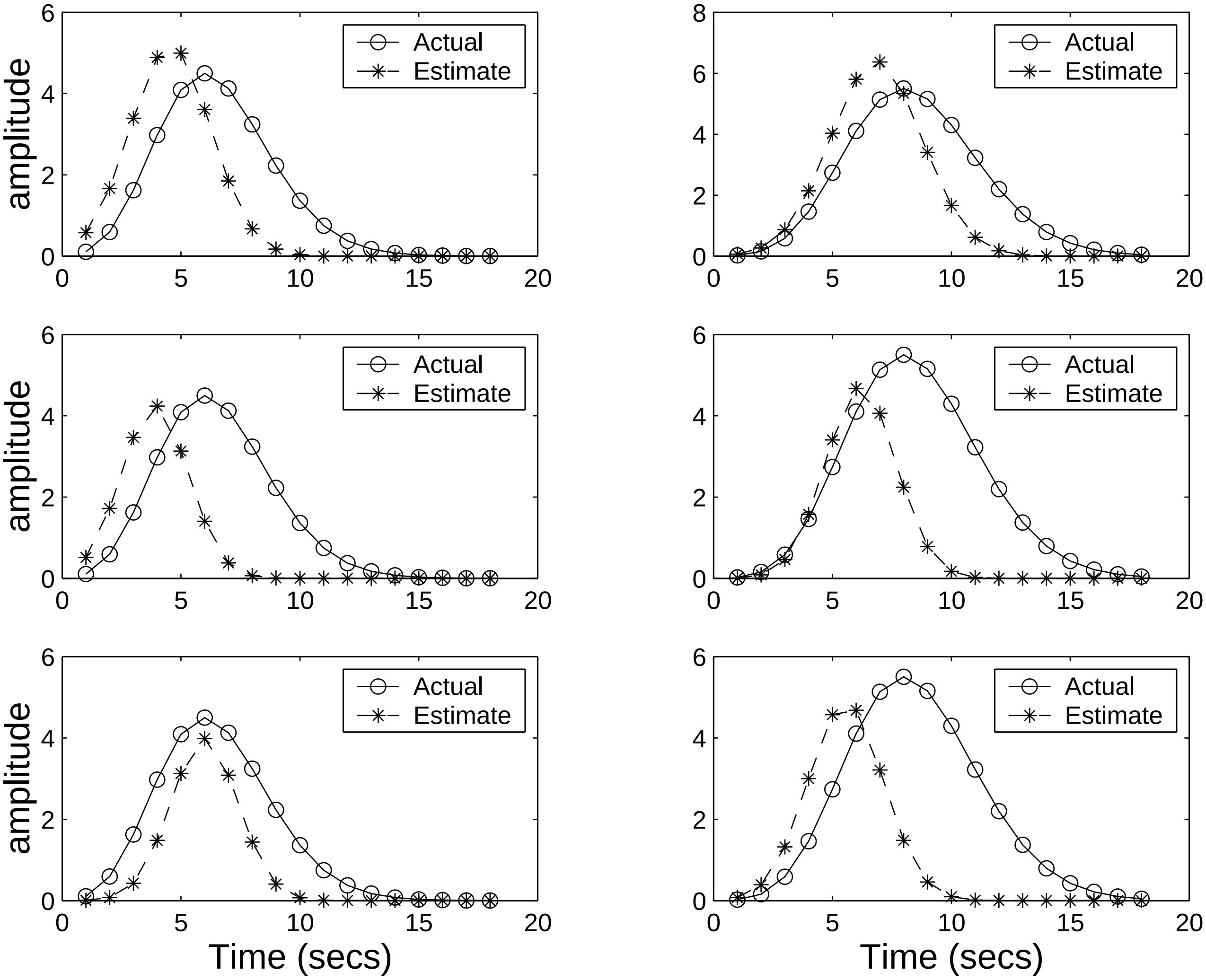

Figure 4.16: Estimation of HRFs of Classification and Estimation algorithm for the Poisson simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. Left panel shows estimation performance for $h1$ and the right panel for $h2$ at $TR = 1, 2, 3$ seconds.

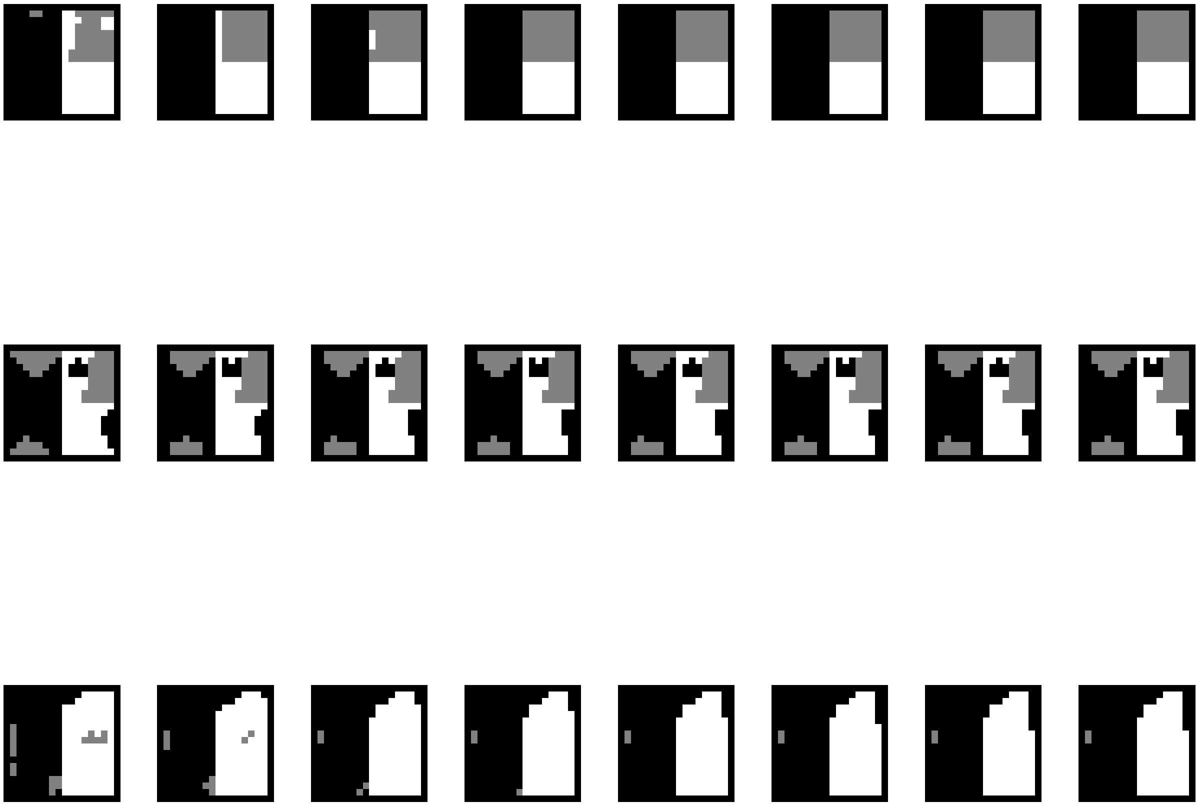

Figure 4.17: Convergence of Classification and Estimation algorithm for the Gamma simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. The algorithm is able to classify the voxels into three classes for $TR = 1$ second (Row(1)). But for $TRs = 2, 3$ seconds, the algorithm is unable to classify (Row(2) and Row(3)) the voxels.

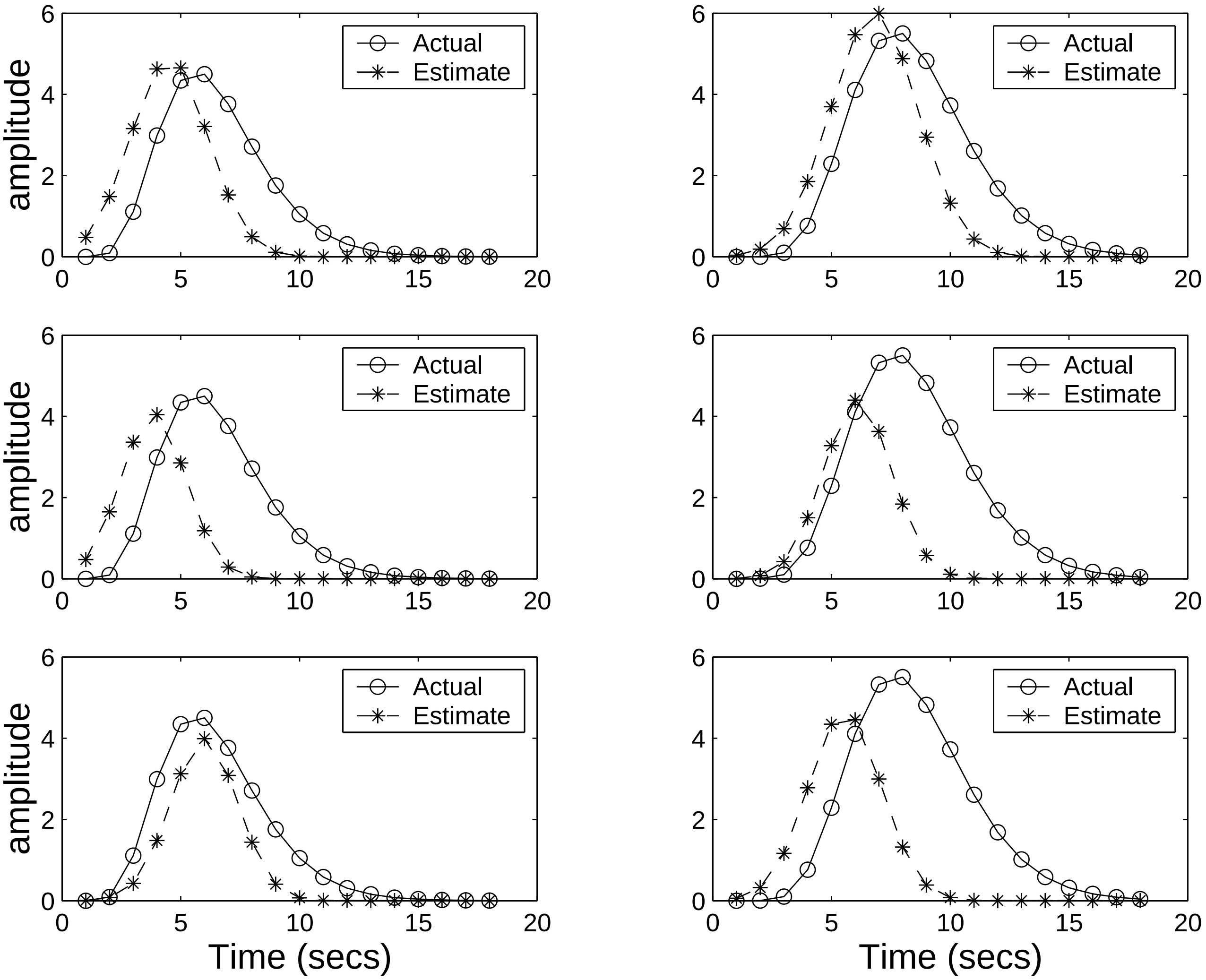

Figure 4.18: Estimation of HRFs of Classification and Estimation algorithm for the Gamma simulated HRfs at sampling intervals: Row (1) $TR = 1$ sec, Row(2) $TR = 2$ sec and Row(3) $TR = 3$ sec. Left panel shows estimation performance for $h1$ and the right panel for $h2$ at $TR = 1, 2, 3$ seconds.

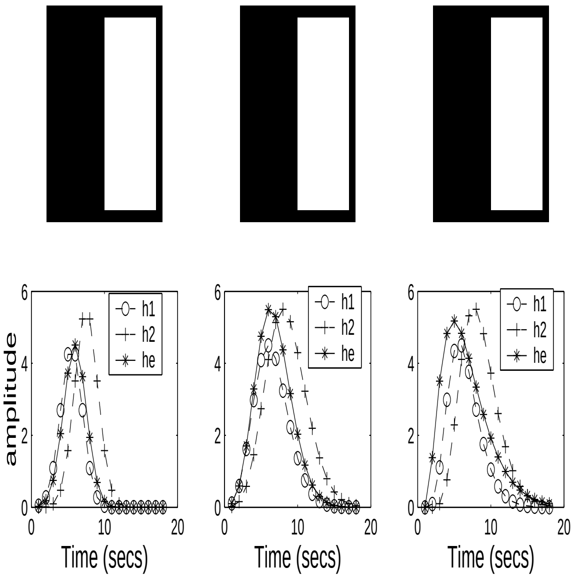

Figure 4.19: Estimation and Classification Performance assuming number of classes to be 2, whereas actual number of classes is 3. The distinction between class 2 and class 3 is lost. Row (1) shows classification performance for gaussian, poisson and gamma HRFs. Row(2) shows estimation of HRF (he) and actual HRFs ($h1, h2$) for gaussian, poisson and gamma HRFs. These simulations are obtained at $TR = 1$ sec.

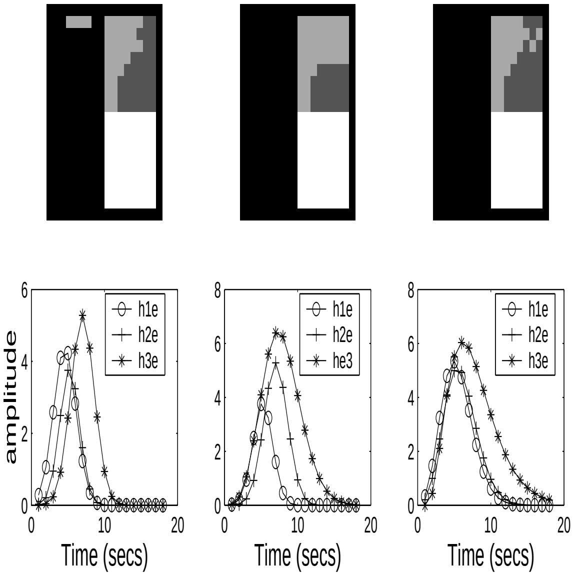

Figure 4.20: Estimation and Classification Performance assuming number of classes to be 4, whereas actual number of classes is 3. Class 2 is divided into different classes. Row (1) shows classification performance for gaussian, poisson and gamma HRFs. Row(2) shows estimation of HRF ($h1e, h2e, h3e$) for gaussian, poisson and gamma HRFs. These simulations are obtained at $TR = 1$ sec.

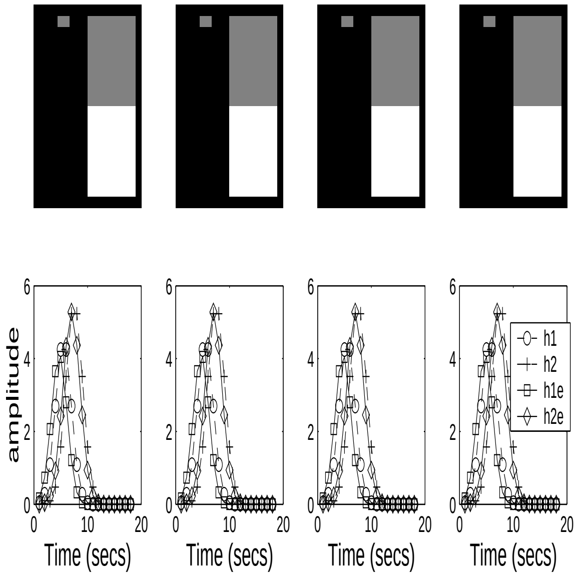

Figure 4.21: Estimation and Classification Performance at $\beta = 0.5, 1, 1.5, 2$ for gaussian HRF model. Performance do not change with change in β value. Row (1) shows classification performance at above βs. Row(2) shows estimation of HRFs. These simulations are obtained at $TR = 1$ sec.

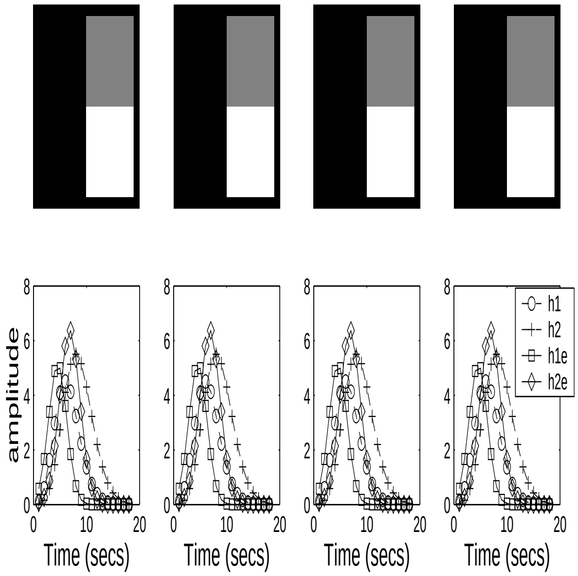

Figure 4.22: Estimation and Classification Performance at $\beta = 0.5, 1, 1.5, 2$ for poisson HRF model. Performance do not change with change in β value. Row (1) shows classification performance at above βs. Row(2) shows estimation of HRFs. These simulations are obtained at $TR = 1$ sec.

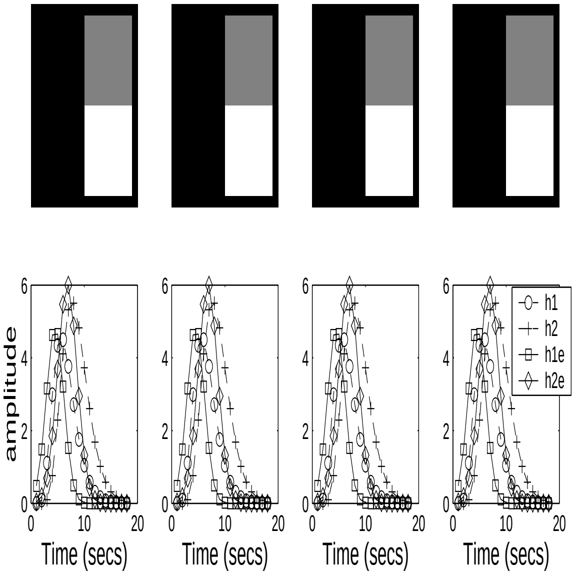

Figure 4.23: Estimation and Classification Performance at $\beta = 0.5, 1, 1.5, 2$ for gamma HRF model. Performance do not change with change in β value. Row (1) shows classification performance at above βs. Row(2) shows estimation of HRFs. These simulations are obtained at $TR = 1$ sec.

Chapter 5

fMRI Time-series modeling by fBm

5.1 Introduction

In the last chapter, we modeled fMRI time series using PPCA which assumes fMRI time series to be a stationary process. However, it is observed that fMRI time series, in the absence of experimental conditions, exhibits $1/f$-like behaviour with disproportionate spectral power at low frequencies. Many physical processes demonstrate this kind of behaviour. These processes are nonstationary and can be conveniently modeled by Fractional Brownian Motion (fBm). It has been observed that [62] wavelet transform (WT) with a sufficient number of vanishing moments decorrelates fBm processes. i.e., WT approximately diagonalizes the covariance matrix of a fBm process. It has further been shown that wavelet coefficients are stationary and uncorrelated within and across the scales even if the original process is nonstationary. Hence WT can be conveniently used to model fBm processes. Wavelet based models have been used in [55], [59] and [60] to model fMRI noise in the context of activation detection.

In our work, we use fBm model for HRF estimation. We model the HRF by both gaussian function and smooth FIR filter as in the previous chapter. The parameters of the models are estimated at each voxel using Maximum a Posteriori Estimation (MAP). We study the influence of sampling time (TR) and shape assumption on the estimation performance. The baseline drifts are also estimated along with the HRF. We compare estimation results with that of PPCA model and the method of [45].

5.2 Fractional Brownian Motion

fMRI voxel time series exhibits serial dependence even in the absence of experimental effects. This coloured fMRI noise typically has disproportionate spectral power at low frequencies, i.e., its spectrum has $1/f$-like behaviour. Physical systems in which many particles are relaxing from excited states at different rates are well known generators of $1/f$-like noise. The colouring in fMRI noise in the absence of physiological effects can be attributed to physical effects [73]. Fractional Brownian Motion (fBm) is a convenient tool to model these processes. In the following, we give a brief review of fBm processes. A detailed treatment can be obtained from [62] and [?].

A fBm $B(t)$ is a zero mean, nonstationary, and non-differentiable function of time. The mean square difference in value of the function between any two time points is proportional to the time difference Δ raised to the power of twice the Hurst exponent H. For fBm processes, H has value between 0 and 1; i.e.,

$$E(B(t) - B(t-\Delta)^2) \approx |\Delta|^{2H} \tag{5.1}$$

If $\Delta = 0$, the variance of the process is nonstationary

$$E(B(t)^2) \approx t^{2H} \tag{5.2}$$

The covariance between the process at two times t and u is given by,

$$\begin{aligned} E(B(t), B(u)) &= 0.5\sigma^2(|t|^{2H} + |u|^{2H} - |t-u|^{2H}) \\ 0.5\sigma^2 &= \Gamma(1-2H)cos(\pi H)/\pi H \end{aligned} \tag{5.3}$$

From 5.3, it can be shown that $B(t)$ is self-affine, i.e., rescaling the process in time by an arbitrary scalar $s > 0$ yields a process $B(st)$ with identical statistical properties to the process $s^H B(t)$ generated by rescaling its values on the original time scale by the scalar s^H. From this self-affine property, it follows that the fractional dimension D of the process is simply related to its Hurst component by

$$D = T + 1 - H \tag{5.4}$$

where T is the topological dimension of the data. For a time series, $T = 1$; therefore $D = 2 - H$. Ordinary Brownian motion is the special case of fBm occurring when $H = 0.5$.

In [61], [74], it was shown that wavelet transform with a sufficient number of vanishing moments decorrelates a fBm process. This means that the wavelet coefficients across and within scales are uncorrelated even if the signal is correlated in the time domain. It was further shown in [56], [62] that wavelet decomposition also possesses optimally decorrelating or Karhunen-Loeve properties for the wider class of $1/f$-like signals. For fBm processes, the expected correlation between any two wavelet coefficients $w_{j,k}$ and $w_{j',k'}$ is (first index refers to scale and the second to time)

$$E(w_{j,k}, w_{j',k'}) \propto O(|2^j k - 2^{j'} k'|^{2(H-R)} \tag{5.5}$$

Thus, the expected correlation between any two wavelet coefficients at the same scale $w_{j,k}$ and $w_{j,k'}$ is

$$E(w_{j,k}, w_{j',k'}) \propto O(|k - k'|^{2(H-R)} \tag{5.6}$$

where R is the number of vanishing moments of the mother wavelet. These results imply that the second order stochastic properties of wavelet coefficients are stationary within and across the scales, even if the original time series is nonstationary. Since the Hurst component $H < 1$ and if $R \geq 2$, the correlation between any two pairs of wavelet coefficients decays rapidly as inverse square of their separation within and across scales. More precisely, it is shown in [75] that provided $R > 2H+1$, the inter coefficient correlations decay hyperbolically fast within levels and exponentially fast between levels. This suggests that the minimum number of vanishing moments required to decorrelate fBm process with $1 < H < 1$ is 4. Wavelet filters with large number of vanishing moments have longer filter lengths. Since fMRI time series typically have shorter lengths, boundary correction results in artefactual inter coefficient correlations. For this reason, we use fourth-order daubechies wavelet which is the most compactly supported wavelet with four vanishing moments.

We use the decorrelating property of wavelet transforms to build a probability model for fMRI time series. This model in turn is used to estimate the hemodynamic response function in the wavelet domain.

5.3 Probability Model for fMRI Time-Series

The BOLD response of the brain for a given task can be modeled as convolution of the HR function and the input task [42]. The input task $x(t)$ is considered as a binary function of time which has a value 1 during the period of task and a 0 during the rest period.

$$y_m(t) = \sum_{k=0}^{K} x(t-k)h_m(k) + \beta_m(t) + e_m(t), \quad t = (K+1)....N \tag{5.7}$$

where, $y_m(t)$ is the observed m^{th} voxel time-series, $h_m(t)$ is the system impulse response, $\beta_m(t)$ is the trend, $e_m(t)$ is the additive noise component and K is the length of the system response. We model the system response $\mathbf{h_m}$ by both gaussian function and smooth FIR filter as discussed in the previous chapter. For convenience, we omit the subscript "m" denoting HRF for each voxel time-series. Note that HRF is computed separately at each voxel. Also, the noise components $\mathbf{e}$ and β are modeled separately for each voxel time-series. The above equation can be written in matrix form as:

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \beta + \mathbf{e} \tag{5.8}$$

where X is a $(N - K \times K)$ convolution matrix and $\mathbf{y}$, β, $\mathbf{e}$ are $N - K \times 1$ vectors. We model the noise component $\mathbf{e}$ as a fBm process. Therefore, the probability of $\mathbf{y}$ given $\mathbf{h}$ is

$$\mathbf{y}|\mathbf{h} \sim \mathbf{N}(\mathbf{Xh} + \beta, \mathbf{C}) \tag{5.9}$$

where C is the covariance matrix whose elements are given by (5.3). Multiplying WT matrix W_T (or applying discrete wavelet transform) on both sides of (5.8)

$$W_T\mathbf{y} = \mathbf{W_T X h} + \mathbf{W_T}\beta + \mathbf{W_T e} \tag{5.10}$$

where, $W_T\mathbf{y}$, $W_T\beta$ and $W_T\mathbf{e}$ are, respectively, the wavelet coefficients of the observed signal, drift and noise. Let $\mathbf{y_w} = \mathbf{W_T y}$, $X_w = W_T X$, $\beta_\mathbf{w} = \mathbf{W_T}\beta$ and $\mathbf{e_w} = \mathbf{W_T e}$. Then the above equation can be written compactly as

$$\mathbf{y_w} = \mathbf{X_w h} + \beta_\mathbf{w} + \mathbf{e_w} \tag{5.11}$$

For example, for J scales ($J = log_2 N$, where N is length of the vector), $\mathbf{y_w}$ is

$$\mathbf{y_w} = [\mathbf{ay_1^J}, \mathbf{dy_1^J}, \mathbf{dy_1^{J-1}}, .., \mathbf{dy^{J-1}_{2^{-J}N}}, ..., \mathbf{dy_1^1}, .., \mathbf{dy^1_{2^{-1}N}}]^\mathbf{t} \tag{5.12}$$

where, ay_k^j and dy_k^j (subscript k denotes time index and superscript j indicates scale) are approximation and detail coefficients respectively. X_w can be obtained by applying wavelet transform to each column of X. Now, the probability model for $\mathbf{y_w}$ given $\mathbf{h}$ is

$$\mathbf{y_w}|\mathbf{h} \sim \mathbf{N}(\mathbf{X_w h} + \beta_\mathbf{w}, \mathbf{\Lambda}) \tag{5.13}$$

where, Λ is a diagonal matrix assuming that WT decorrelates the noise process $\mathbf{w}$. The diagonal elements of Λ are variances of wavelet coefficients at each scale.

$$\Lambda = diag[\sigma_J^2, \sigma_J^2, ..., \sigma_1^2] \tag{5.14}$$

5.4 Estimation of HRF and drift

5.4.1 Gaussian Model for HRF

In this section, we model HRF as a gaussian function which is described in the previous chapter. The unknown parameters to be estimated are parameters of HRF θ and the drift component β. The origin of drift in fMRI time-series is not known. But it is observed that it is a low frequency signal. In this work, we estimate the drift in wavelet domain using the information that it is a low frequency signal. Therefore, its wavelet coefficients at lower scales (high frequencies) should be negligible and can be assumed to be zero. Hence the wavelet coefficients for the drift component can be written as:

$$\begin{aligned}\beta_w = [&ay_1^J, dy_1^J, dy_1^{J-1}, .., dy_{2^{-J}N}^{J-1}, \\ &., dy_1^{Jo}, .., dy_{2^{-Jo}N}^{Jo}, 0, 0......0]'\end{aligned} \tag{5.15}$$

where, Jo is the lowest scale upto which the drift component is significant. $a\beta_k^j$ and $d\beta_k^j$ are, respectively, approximation and detail coefficients of $\beta_\mathbf{w}$. We observed from the null data of fMRI that the drift component is significant only at first two higher scales. Therefore we assume $Jo = J - 2$. We use the following iterative algorithm to estimate the unknown parameters:

1. Detrend the voxel time series using the procedure described in the initialization step for the covariance matrix estimation in the previous chapter. Find a least square estimate of $\mathbf{h}$

   $$\hat{\mathbf{h}}_\mathbf{l}\mathbf{s} = (X_w' X_w)^{-1} X_w' \mathbf{y}_{wd} \tag{5.16}$$

   where, $\mathbf{y_{wd}}$ is the wavelet coefficients of the detrended voxel time series.

2. Remove the signal component from the time-series

   $$\tilde{\mathbf{y}}_\mathbf{w} = \mathbf{y}_\mathbf{w} - \mathbf{X}_\mathbf{w}\hat{\mathbf{h}}_\mathbf{ls} \tag{5.17}$$

3. Estimate the drift component $\hat{\beta}_\mathbf{w}$ by equating the wavelet coefficients to zero from

scale $Jo - 1$ to 1.

4. Now, use the estimated drift in the wavelet domain and find the MAP estimate of θ as:

$$\begin{aligned} \theta_{MAP} &= arg\ max_\theta\ p(\mathbf{y}_w|\mathbf{h})p(\theta) \\ &= arg\ min_\theta\ (\mathbf{y}_w - X_w\mathbf{h}(\theta) - \hat{\beta}_w)^t(\Lambda)^{-1} \\ (& \mathbf{y}_w - X_w\mathbf{h}(\theta) - \hat{\beta}_w) + \theta^t V_\theta^{-1}\theta \end{aligned} \tag{5.18}$$

   Where, the prior probability $p(\theta) \ \sim N(\mathbf{m}_\theta, V_\theta)$ is defined in the previous chapter. Let $\hat{\mathbf{h}}_{MAP} = \mathbf{h}(\theta_{MAP})$.

5. Repeat steps (2)-(4) by replacing $\hat{\mathbf{h}}_{\mathbf{l}}\mathbf{s}$ by $\hat{\mathbf{h}}_{MAP}$ until convergence.

This alogorithm converges in 2 to 3 iterations. The above algorithm is applied at each voxel time-series.

5.4.2 Smooth FIR filter model for HRF

In this section, we model the HRF by a smooth FIR filter instead of the gaussian model. The approach is the same as in the previous chapter except that the noise component is modeled by fBm process instead of PPCA model.

The observed time series model for a voxel with the wavelet transformed drift component $\beta_{\mathbf{w}}$ estimated and removed is given by:

$$\mathbf{y}_w = X_w\mathbf{h} + \mathbf{e}_w \tag{5.19}$$

where, $\mathbf{h}$ is modeled by a smooth FIR filter. Now, the pdf of $\mathbf{y}_w$ given $\mathbf{h}$ is

$$\mathbf{y}_w|\mathbf{h} \sim N(X_w\mathbf{h}, \Lambda) \tag{5.20}$$

where, Λ is a diagonal matrix given by 5.14. Following the same procedure as in the previous chapter, the MMSE estimate of $\mathbf{h}$ is given by the mean of the posterior distribution of $\mathbf{h}$ given the data $\mathbf{y}_w$. The MMSE estimate and covariance matrix of $\mathbf{h}$ are given by

$$\begin{aligned} E(\mathbf{h}|\mathbf{y}_w, \hat{\epsilon}) &= (X_w^t \Lambda^{-1} X_w + \hat{\epsilon}^2 M_p)^{-1} X_w^t \Lambda^{-1} \mathbf{y}_w & (5.21) \\ C_{\frac{h}{y}} &= (X_w^t \Lambda^{-1} X_w + \hat{\epsilon}^2 M_p)^{-1} & (5.22) \end{aligned}$$

The factor $\hat{\epsilon}^2$ determines the relative weight of the prior information. We get a Maximum Likelihood Estimate for $\hat{\epsilon}^2 = 0$. $\hat{\epsilon}$, the MAP estimate of ϵ is given by:

$$\hat{\epsilon} = arg_\epsilon \ max \ p(\epsilon|\mathbf{y}_w) \tag{5.23}$$

The marginal posterior probability for ϵ is given by:

$$\begin{aligned} p(\epsilon|\mathbf{y_w}) &\propto (det(M(\epsilon))^{-1/2} \epsilon^{\frac{(K-1)}{2}} exp(-1/2 \mathbf{y}_w^t \Lambda^{-1} \mathbf{y}_w - \hat{\mu_B}^t M(\epsilon) \hat{\mu_B}) & (5.24) \\ where, \quad M(\epsilon) &= X_w^t \Lambda^{-1} X + \epsilon^2 M_p \\ and, \quad \hat{\mu_B} &= M(\epsilon)^{-1} (X_w^t \Lambda^{-1} \mathbf{y}_w) \end{aligned}$$

The algorithm given in the previous section can be used to estimate the HRF $\mathbf{h}$ and the drift component β by replacing $\hat{\mathbf{h}}_{MAP}$ in that algorithm by the MMSE estimate of $\mathbf{h}$ given by 5.21.

5.4.3 Estimation of the Covariance Matrix

We need the covariance matrix Λ for the above algorithm. The diagonal elements of Λ are nothing but the variances of wavelet coefficients at each scale. There are two ways in which one can estimate the diagonal elements of Λ. The variance of wavelet coefficients at the scale j for the fBm process is related to Hurst component H as [62]:

$$\sigma_j^2 = \sigma_b^2 2^{-(2H+1)j} \tag{5.25}$$

where σ_b is a proportionality constant. The variance of wavelet coefficients can be estimated by estimating H and σ_b. There are a large number of algorithms in the literature [62] which give maximum likelihood estimates of these parameters. These algorithms are not only computationally expensive but also not reliable for $H > 0$ [76]. Another way of estimating the variance at each scale is to use simple sample variance estimator [77]. fMRI time-series are typically of length 128 or less. The length of wavelet coefficients at scale j will be $N/2^j$. Hence, sample variance estimator will be unreliable due to small data lengths. In [77], it is shown that variance estimators will be more reliable in undecimated wavelet transforms (UWT) compared to the conventional non-redundant discrete wavelet transform (DWT). In UWT, the length of wavelet coefficients will be N irrespective of the scale. In this work, we use DWT for estimating the unknown parameters θ and β and UWT for estimating the variances at each scale.

5.5 Results and Discussion

We test the performance of the above two algorithms on synthetic and real (Null) fMRI data using the same performance measures described in the last chapter. We also study the influence of sampling time (TR) and shape of HRF on the estimation performance. We use the same procedure for BOLD and noise fMRI simulation described in the previous chapter.

5.5.1 Synthetic Data

Tables 5.1 and 5.2 summarize the performance of the parametric and FIR filter methods at different SNRs, TRs and HRF functions. The parametric method gives upta $15 - 20$ error each in the estimation of time to peak and amplitude, respectively, at $TR = 1$ for all HRF models and SNRs. The performance is robust for decrease in SNR. The sample correlation is more than 0.8. In the case of FIR model, the estimation error is about $3-8\%$ and $7-18\%$ in the estimation of time to peak and amplitude, respectively, at $TR = 1$ for all HRF models and SNRs. The sample correlation is more than 0.9. Hence, FIR filter

Table 5.1: Performance Comparison of Parametric method at different SNRs and TRs. Poisson,Gamma and Gaussian simulated Hemodynamic Response Functions are modeled by a Gaussian Function. fMRI noise is modeled by fractional Brownian motion. poi: Poisson, gam: Gamma, gau: Gaussian, δt_p: percent error in time to peak, $\delta\eta$: percent error in amplitude, MSE: Mean square error in HRF estimation and ρ: sample correlation between actual and estimated HRFs.

SNR	TR	δt_p			$\delta\eta$			MSE			ρ		
(dB)	(sec)	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
5	1	19	20	16.4	12	11	13.2	1.95	1.8	1.12	0.85	0.83	0.84
5	2	26	28	24	22	24.7	24.2	2.3	2.5	1.96	0.6	0.5	0.62
5	3	26	27	28	28	27.4	29	2.76	3	2.98	0.55	0.48	0.46
0	1	18.6	20.3	16.5	17	16	18.58	1.96	1.88	1.31	0.81	0.82	0.81
0	2	20	21	18.2	25	24	25.35	2.1	2.11	2.4	0.63	0.68	0.61
0	3	26	27	29	23	22.1	23.13	3.35	3.1	2.9	0.53	0 .49	0.48
-5	1	18	20	16.6	21	20	19	2	1.96	1.4	0.82	0.81	0.81
-5	2	23	25	27	24	27	26	2.73	2.88	2.9	0.62	0.64	0.65
-5	3	31	32	33	30	33	34	3.4	3.5	3.2	0.46	0.45	0.43
-10	1	17	18	15	18	16.4	15	1.76	1.73	1.4	0.81	0.79	0.78
-10	2	24	23	24	27	28	24	2.75	2.9	2.82	0.55	0.58	0.59
-10	3	33.7	34	32	32	33.2	34	3.7	3.5	3.4	0.47	0.46	0.43

performance is better than the gaussian model for $TR = 1$. However, at $TR = 2$, the performance of FIR filter model is poorer than the parametric method. Also, performance decreases with the decrease in SNR. At $TR = 3$, the estimation of HRF becomes very unreliable. These results are comparable to that obtained by PPCA model of the previous chapter.

5.5.2 Null Data

Now, we apply both parametric and nonparametric methods on the Null fMRI data. Figures 5.1 and 5.2 respectively show the performance of parametric and FIR modeling methods at TR = 1, 2 and 3 seconds. At TR = 1 sec, the error in the estimation accuracy of time to peak and amplitude is about $14 - 18\%$ and $12 - 14\%$ respectively for all the three simulated HRFs for the parametric method. FIR filter method is able to estimate

Table 5.2: Performance Comparison of nonparametric method at different SNRs and TRs. Poisson,Gamma and Gaussian simulated Hemodynamic Response Functions are modeled by a Smooth FIR filter. fMRI noise is modeled by fractional Brownian motion. poi: Poisson, gam: Gamma, gau: Gaussian.

SNR	TR	δt_p			$\delta\eta$			MSE			ρ		
(dB)	(sec)	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
5	1	3	2.7	2.3	7.2	12	14	0.38	0.35	0.43	0.97	0.96	0.95
5	2	26	27	23.4	26	29	36	2.9	2.77	2.84	0.51	0.52	0.54
5	3	33	31	27	29	31	32	3.4	3.8	3.7	0.44	0.42	0.49
0	1	4.1	4.6	5	7.4	15.5	15	0.8	0.8	0.84	0.93	0.93	0.9
0	2	28	29	25.4	28	30	37	3.1	3.2	3.3	0.53	0.52	0.51
0	3	31	37	42	34	35	31	4.2	4.3	4.4	0.38	0.39	0.41
-5	1	5.2	5.5	5.3	7.7	17	18	1.5	1.6	1.72	0.87	0.85	0.82
-5	2	32	33	39	27	29	30	3.7	3.6	3.5	0.46	0.45	0.47
-5	3	41	49	66	34	37	35	6.5	6.4	6.2	0.3	0.35	0.34
-10	1	6.2	7.5	8	12	17.2	17.1	3.28	3.38	3.4	0.8	0.74	0.73
-10	2	-	-	-	-	-	-	-	-	-	-	-	-
-10	3	-	-	-	-	-	-	-	-	-	-	-	-

these parameters with accuracy 3% and 10%, respectively. But at higher TRs, the ability of FIR filter to model HRFs goes down as compared to parametric model which is also evident from the simulated data. The sample correlation ρ between the estimated HRF and actual HRF is about 0.82 for parametric case and is about 0.9 for nonparametric model. Hence at low TRs, both parametric and FIR filter models are able to recover HRFs with good accuracy. At TR = 2 sec, the error in the estimation accuracy of time to peak and amplitude is about 18% and 17%, respectively, for all the three simulated HRFs for the parametric method, whereas FIR filter method is able to estimate these parameters with accuracy of 25% and 17%, respectively. The sample correlation for parametric case is about 0.7 and for FIR case, it is only 0.5. Hence parametric model performs better at higher TRs. These results are also comparable to that of previous chapter obtained using PPCA modeling of fMRI time series.

5.5.3 Overall Comparison

We compare the performance of PPCA and fBm noise models for the smooth FIR filter method with the method of [45]. In [45], the drift component is modeled by polynomial bases and the noise component by white gaussian noise. The model is given by

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{P}\lambda + \mathbf{e} \tag{5.26}$$

where $\mathbf{y}$ is the observed time series, X is the convolution matrix, the columns of P represent polynomial bases of degree 2, λ are the unknown coordinates of drfit component along the polynomial bases and $\mathbf{e}$ is white gaussian noise with unknown variance. The HRF $\mathbf{h}$ is modeled by smooth FIR filter. We apply this method on both synthetic and Null fMRI data described in the previous chapter. Table 5.3 summarizes the performance of this method on the synthetic data at different SNRs and TRs. By comparing this table with tables 4.2 and 5.2, the following conclusions can be drawn. The performance of PPCA and fBm models is superior to the polynomial model of [45]. For example, at $SNR = 0$ dB and $TR = 1$ second, the estimation error in time to peak δt_p is about 3% for PPCA model and 4.5% for fBm model for all the simulated HRF parametric models (gaussian, poisson and gamma). Whereas, polynomial noise model gives about 10% error. The mean square error (MSE), 0.7 and 0.8, respectively for PPCA and fBm models, is less compared to 1.2 by the polynomial model. At this SNR and TR, the correlation coefficient ρ, 0.9, obtained by PPCA and fBm models is superior to 0.8, achieved by the polynomial model. Performance of all the three models degrades with the decrease in SNR and increase in TR. The degradation in performance is more severe in the polynomial model. Table 5.4 summarizes the results obtained at SNR = 0 dB and $TR = 1$ second for all the three models.

In the case of Null data, for $TR = 1$ second, PPCA and fBm models are able to estimate time to peak with an accuracy of 4% and 3%, respectively . Polynomial model gives an accuracy of about 7%. At this TR, MSEs for PPCA and fBm models are respectively about 0.8 and 0.9, less than 1.1 achieved by the polynomial model. The

Table 5.3: Performance Comparison of nonparametric method at different SNRs and TRs. Poisson,Gamma and Gaussian simulated Hemodynamic Response Functions are modeled by a Smooth FIR filter. Trends are modeled by polynomial bases upto degree 2. poi: Poisson, gam: Gamma, gau: Gaussian, δt_p: percent error in time to peak, $\delta\eta$: percent error in amplitude, MSE: Mean square error in HRF estimation and ρ: sample correlation between actual and estimated HRFs.

SNR	TR	δt_p			$\delta\eta$			MSE			ρ		
(dB)	(sec)	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
5	1	8.54	8.61	11.3	8.5	12.7	21	0.87	1	1.3	0.91	0.88	0.81
5	2	24	25	24	25	29.3	29	2.97	2.4	2.73	0.51	0.52	0.56
5	3	26	27	30	31	32	31	3.67	3.44	3.7	0.38	0.39	0.4
0	1	8.7	9.2	12	6.4	9.4	19	1	1.2	1.5	0.89	0.85	0.8
0	2	20	21	25	27	28.2	27.3	2.3	2.6	3	0.51	0.55	0.53
0	3	25	28	31	32	31.6	34	3.8	4	4.7	0.4	0.39	0.38
-5	1	10	11.4	11	14	12	11	2	2.2	2.4	0.8	0.8	0.7
-5	2	20	17	21	29	28	27	4.5	4	4.6	0.47	0.44	0.49
-5	3	31	35	33	34	33	31	13	9	7	0.31	0.34	0.4
-10	1	12	8	24	14	11	12	5	5	5.6	0.7	0.65	0.55
-10	2	-	-	-	-	-	-	-	-	-	-	-	-
-10	3	-	-	-	-	-	-	-	-	-	-	-	-

fidelity index ρ obtained by PPCA and fBm models is about 0.92 is superior to 0.85 achieved by the polynomial model. Figure 5.3 shows the performance of polynomial bases model for HRF estimation. For all the three models, HRF estimation becomes unreliable with increase in TR. From this, we can conclude that PPCA and fBm noise models are superior to the adhoc polynomial bases, for modeling the baseline drifts and fMR noise. Table 5.5 summarizes the results obtained at $TR = 1$ second on the null data for all the three models.

5.6 Conclusions

We proposed wavelet based methods for the estimation of hemodynamic response function for both gaussian and FIR filter models. The fMRI noise which exibhits $1/f$-like spectrum is modeled as fractional Brownian Motion. The probability models are built

Table 5.4: Comparison of fMRI models on synthetic data at $SNR = 0$ dB and $TR = 1$ sec. HRF is modeled by Smooth FIR filter.

Methods	δt_p			$\delta \eta$			MSE			ρ		
	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
PPCA	3.52	3.12	3.12	7.0	5.65	13	0.42	0.52	0.72	0.95	0.93	0.9
fBm	4.1	4.6	5	7.4	15.5	15	0.8	0.8	0.84	0.93	0.93	0.9
Poly. Bases	8.7	9.2	12	6.4	9.4	19	1	1.2	1.5	0.89	0.85	0.8

Table 5.5: Comparison of fMRI models on Null data at $TR = 1$ sec. HRF is modeled by Smooth FIR filter.

Methods	δt_p			$\delta \eta$			MSE			ρ		
	poi	gam	gau	poi	gam	gau	poi	gam	gau	poi	gam	gau
PPCA	3.1	5	5.7	6.8	8	3	0.7	0.8	0.8	0.93	0.91	0.91
fBm	2	3	3	10	8	12.4	0.9	0.97	1.15	0.93	0.92	0.9
Poly. Bases	6.64	6.4	8.35	8.6	13	21	0.9	1.0	1.24	0.9	0.88	0.81

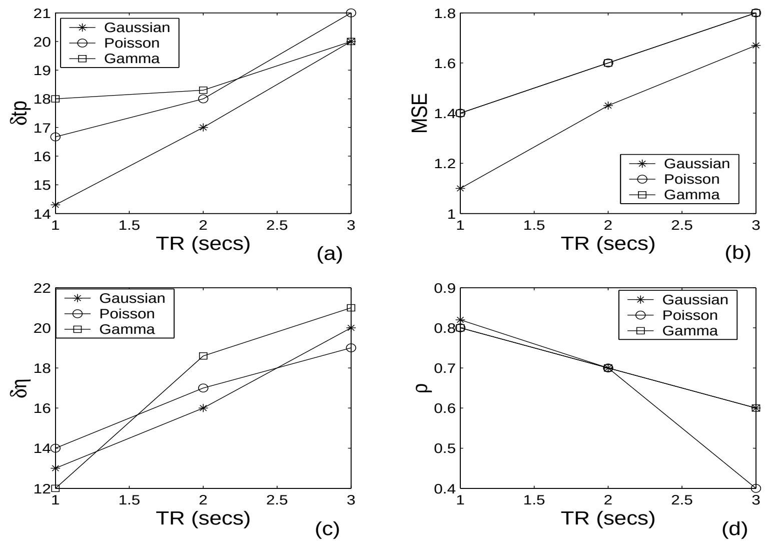

Figure 5.1: Performance measures for parametric modeling of HRF at TR = 1, 2 and 3 secs. fMRI noise is modeled by fractional Brownian motion. (a) Percentage error in the estimation of time to peak ($\delta t_p \times 100$) (b) MSE vs TR (c) Percentage error in the estimation of amplitude of HRF ($\delta \eta \times 100$) vs TR (d) Correlation between estimated and actual HRF (ρ).

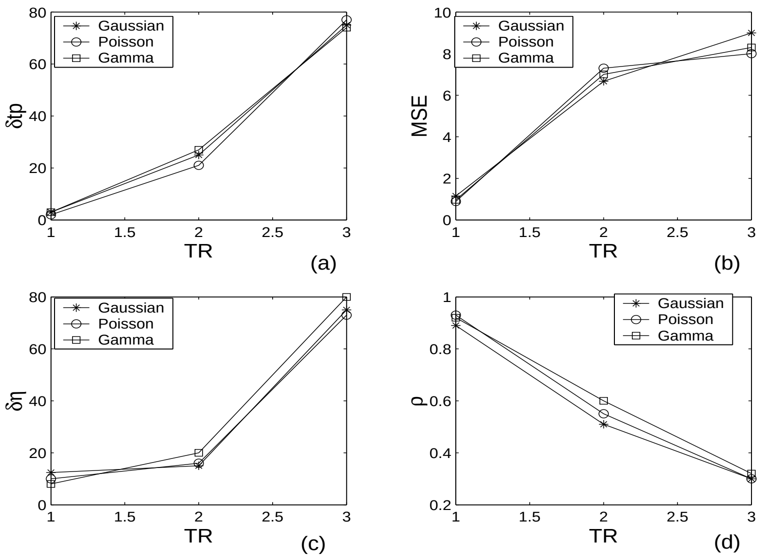

Figure 5.2: Performance measures for nonparametric modeling of HRF at TR = 1, 2 and 3 secs. fMRI noise is modeled by fractional Brownian motion. (a) Percentage error in the estimation of time to peak ($\delta t_p \times 100$) (b) MSE vs TR (c) Percentage error in the estimation of amplitude of HRF ($\delta\eta \times 100$) vs TR (d) Correlation between estimated and actual HRF (ρ).

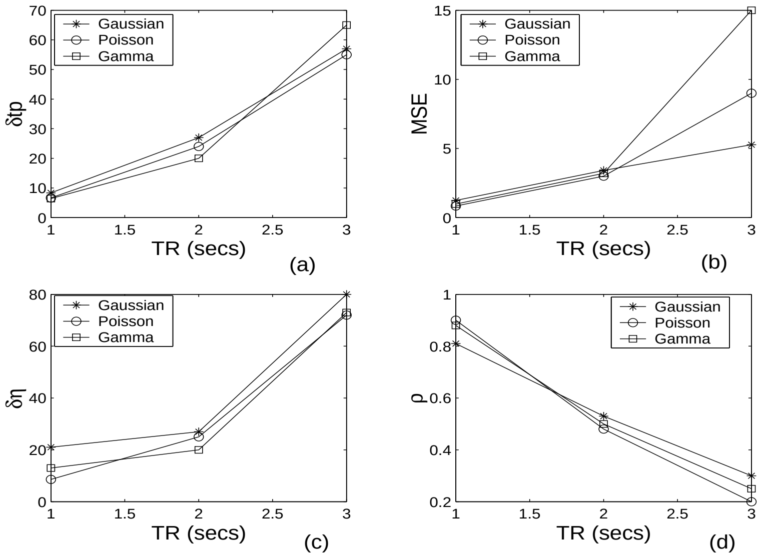

Figure 5.3: Performance measures for nonparametric modeling of HRF at TR = 1, 2 and 3 secs. Trends are modeled by polynomial bases upto degree 2.(a) Percentage error in the estimation of time to peak ($\delta t_p \times 100$) (b) MSE vs TR (c) Percentage error in the estimation of amplitude of HRF ($\delta\eta \times 100$) vs TR (d) Correlation between estimated and actual HRF (ρ).

on the assumption that wavelet transforms with sufficient number of vanishing moments decorrelates fBm process. The baseline drifts are also estimated along with HRF. The obtained estimation accuracy for both gaussian and smooth FIR filter models is comparable to that obatained by PPCA model. We also compared the PPCA and fBm methods with the exsisting method of [45] which uses polynomial bases for modeling baseline drifts and smooth FIR filter model for HRF. Both PPCA and fBm models using smooth FIR filter model for HRF are found to be superior to the method of [45].

Chapter 6

Conclusions

This thesis mainly deals with two problems, viz., 3D image compression and fMRI data analysis, in medical image processing. In the first part of the thesis, we proposed two schemes for lossless compression of 2D and 3D magnetic resonance images (MRI) and in the second part, we proposed two probability models for fMRI voxel time series, which in turn are used to estimate hemodynamic response function (HRF).

6.1 Main Contributions

The following are the main contributions of the thesis:

- A new nouniform mesh-based interframe coding has been proposed for 3D MR images. This coding scheme is better than the existing uniform mesh-based scheme of [9] in terms of lossless compression performance.

- Context based entropy coding has been devised to code the residues after motion compensation. 3D energy and texture contexts as in [14] are designed for nonuniform mesh-based scheme.

- 2D and 3D wavelet-based coding schemes with region of interest, progressive transmission and lossless coding capabilities have been proposed for MR images.

- Context-based entropy coding has been used to improve the performance of the wavelet-based coding schemes. 2D and 3D contexts for the significance map are constructed using a mutual information based context quantization algorithm.
- A probability model using probabilistic principal component analysis (PPCA) has been proposed to model fMRI time series. Using this model, the correlated noise components (base line trends and fluctuations) are represented in terms of dominant principal components of the fMRI data. Hemodynamic response function is modeled by parametric (guassian) and nonparametric (smooth FIR) filter) models. This probability model is then used to estimate hemodynamic response function.
- A new bootstrap method to calculate covariance matrix has also been given to take care of spatial noise variation in fMR images.
- Using PPCA, we have also developed a classification algorithm which groups voxels with similar HRFs together. This algorithm can be used in event-related studies.
- We have used fractional Brownian motion (fBm) to model fMRI time series which exibhits $1/f$-like spectral properties. A probability model has been built based on the assumption that wavelet transforms with sufficient number of vanishing moments decorrelates fBm process.
- The above probability model has then been used to estimate HRF which is again modeled by parametric and nonparametric models.

6.2 Conclusions

3D MRI data provided by NIMHANS and standard 3D MRI data (MR-MRI) provided by Gloria Menegaz have been compressed by the proposed nonuniform mesh-based and wavelet-based coding schemes. Nonuniform mesh-based coding schemes give lossless compression of about 2 bits per pixel (*bpp*) on NIMHANS data and about 1.9 *bpp* on the standard MR-MRI data set. Uniform mesh-based scheme of [9] give lossless compression of 3.5 *bpp* on NIMHANS data. We improved the lossless bitrate of this uniform mesh-based scheme by using context based entropy coding on the residue images. Using this scheme lossless compression improved to 2.31 *bpp*. Lossless compression performance of 2D and 3D wavelet-based schemes are 2.1 and 2.2 *bpp*, respectively on NIMHANS data set. On the MR-MRI dataset, the lossless compression performance of 3D wavelet scheme is about 1.83 *bpp*. The performance of our mesh and wavelet-based schemes are better than the recent wavelet-based 3D compression schemes like MLZC and 3D-EZW of [1]. The performances of MLZC and 3D-EZW on MR-MRI data set, as reported in [1], are 2.14 and 2.27 *bpp*, respectively.

We applied both PPCA and fBm based HRF estimation methods on simulated and real fMRI data. We also studied the influence of sampling rate (TR) on the accuracy of HRF estimation on both simulated and real fMRI noise data. We also compared the PPCA and fBm methods with the exsisting method of [45] which uses polynomial bases for modeling baseline drifts and smooth FIR filter model for HRF. Both PPCA and fBm models using smooth FIR filter model for HRF are found to be superior to the method of [45]. However, the fBm model for nonstationay fMRI time series did not show much improvement over the stationary PPCA model. Smooth FIR filter model is found to be a good method at high sampling rates and parametric model is a better choice at lower sampling rates.

6.3 Scope for future work

The mesh-based scheme does not provide progressive transmission capability. In uniform mesh-based scheme of [9], a wavelet-based scheme is used on residue images after motion compensation to achieve progressive transmission of images. Similar stratergy can also be used for our proposed noununiform mesh-based scheme. It will also be interesting to compare the progressive transmission of this scheme with that of wavelet-based schemes of chapter 3. Lossless coding of medical images are required before diagnosis. However, after identifying regions of interest by radiologist, lossless compression can be used on those regions and rest of the regions can be compressed with lossy schemes. Such a hybrid schemes can be built on the two proposed schemes. In the wavelet-based scheme, rectangular lattices are used to exploit intraband correlation. Nonuniform lattices depending on the regions in the wavelet domain can be designed using the procedures developed in chapter 2. One can expect a better exploitation of intraband correlation. These are some of the issues that can be addressed as an extension of this work.

The estimated HRFs using PPCA and fBm models can be used to find activated regions in the brain, which is a future scope of this work. It is also interesting to compare this methodology of activation detection using estimated HRFs with that of conventional methods where a fixed HRF is assumed for entire brain for activation detection.

Appendix A

Integer Wavelet Transform

A.1 Integer Wavelet Transform

We need transforms which map integer-valued image data to integer-valued transform coefficients for compression schemes with lossless compression capability. Such schemes are essential for medical images. In [19], [20], a simple way of constructing wavelet transforms mapping integer-integer values using lifting schemes. This scheme also takes care of boundary extensions at the border of the signal and also reduces the computational complexity with respect to classical filter band implementation of [35].

In [20], it was shown that any discrete wavelet transform or two band filtering scheme with finite filters can be decomposed into a finite sequence of simple filtering steps called as lifting steps. This decomposition corresponds to a factorization of the polyphase matrix of the wavelet or subband filters into elementary matrices. These lifting steps can be obtained using eculidean factorization of the polyphase matrix of the filter bank into a sequence of 2×2 upper and lower triangular matrices.

We briefly discuss the philosophy of lifting schemes. More details can be obtained from [20].

A.1.1 Laurent Polynomials

Let h be a finite response filter (FIR) with a finite number of nonzero coefficients. Let k_b and k_e respectively be the smallest and largest integres for which h_{k_b} and h_{k_e} are non-zero. The z-transform of a FIR filter is a Laurent polynomial h(z) given by

$$h(z) = \sum_{k=k_b}^{k_e} h_k z^{-k} \tag{A.1}$$

The degree of a Laurent Polynomial is defined as: $|h| = k_e - k_b$. The length of filter is the degree of the associated Laurent polynomial plus one. The difference between the Laurent polynomial and a regular polynomial can be explained with the following example. The degree of z^p seen as a regular polynomial is p whereas it is zero if the polynomial is viewed as a Laurent polynomial. The set of all Laurent polynomials with real coefficients has a commutative ring structure. The sum and difference of two Laurent polynomials is agian a Laurent polynomial. The product of a Laurent polynomials of degrees l and p is a Laurent polynomial with degree $l + p$. This ring is denoted as $\mathbf{R}[\mathbf{z}, \mathbf{z^{-1}}]$. Within a ring exact division is not possible in general. Let $a(z)$ and $b(z) \neq 0$ be two Laurent polynomials with $|a(z)| \geq |b(z)|$. Then there always exists a Laurent polynomial $q(z)$ (the quotient) with $|q(z)| = |a(z)| - |b(z)|$, and a Laurent polynomial $r(z)$ (the remainder) with $|b(z)| \geq |r(z)|$ so that

$$\begin{aligned} a(z) &= b(z)q(z) + r(z) \\ where \quad q(z) = a(z)/b(z) \quad and \quad r(z) &= a(z)\%b(z) \end{aligned}$$

The division is exact when $b(z)$ is monomial ($|b(z)| = 0$), then $r(z) = 0$. A Laurent Polynomial is invertible if and only if it is a monomial.

Similarly, 2×2 matrices of Laurent polynomials also form a ring, which is denoted by M(2;$\mathbf{R}[\mathbf{z}, \mathbf{z^{-1}}]$). If the determinant of such a matrix is a monomial, then the matrix is invertible. The set of invertible matrices is denoted GL(2;$\mathbf{R}[\mathbf{z}, \mathbf{z^{-1}}]$). A matrix from this

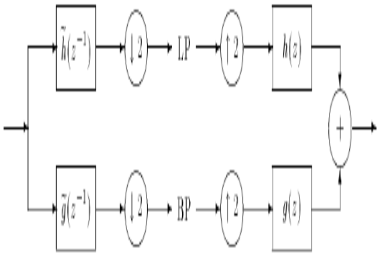

Figure A.1: Block Diagram for two-channel filter bank.

set is unitary (also called as para-unitary) in case

$$M(z)^{-1} = M(z^{-1})^t$$

A.1.2 Two-Channel Filter Bank and Polyphase representation

Discrete Wavelet Transform (DWT) can be implemented by a two-channel filter bank as shown in the figure A.1. The forward transform uses two analysis filters $\tilde{h}$ (lowpass) and $\tilde{g}$ (bandpass) and the inverse transform uses two synthesis filters h (lowpass) and g (bandpass). Let all these filters be FIR filters. The conditions for perfect reconstruction (PR) are

$$h(z)h(\tilde{z}^{-1}) + g(z)g(\tilde{z}^{-1}) = 2$$
$$h(z)h(-\tilde{z}^{-1}) + g(z)g(-\tilde{z}^{-1}) = 0$$

We define a modulation matrix $M(z)$ as

$$M(z) = \begin{bmatrix} h(z) & h(-z) \\ g(z) & g(-z) \end{bmatrix}$$

Similarly, the modulation matrix $\tilde{M}(z)$ can be defined. Now, the PR condition can be written as

$$M(\tilde{z}^{-1})^t M(z) = 2I \qquad \text{(A.2)}$$

where, I is 2×2 identity matrix. If all filters are FIR, then the matrices $M(z)$ and $\tilde{M(z)}$ belong to GL(2;$\mathbf{R}[\mathbf{z}, \mathbf{z^{-1}}]$).

The polyphase representation is a convenient tool to express the special structure of the modulation matrix [**?**]. The polyphase representation of a filter h is given by

$$h(z) = h_e(z) + h_o(z)$$

where, h_e contains the even coefficients and h_o contains the odd coefficients:

$$h_e(z) = \sum_k h_{2k} z^{-k} \quad and \quad h_o(z) = \sum_k h_{2k+1} z^{-k}$$
$$or \quad h_e(z^2) = (h(z) + h(-z))/2 \quad and \quad h_o(z^2) = (h(z) - h(-z))/(2z^{-1})$$

The polyphase matrix $P(z)$ (similarly $\tilde{P(z)}$) is defined as

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$

so that

$$P(z^2)^t = 1/2M(z) \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix}$$

The PR condition is now given by

$$P(z)\tilde{P(z^{-1})}^t = I \tag{A.3}$$

Again we want $P(z)$ and $\tilde{P(z)}$ to contain only Laurent polynomials. Equation A.1.2 implies that $detP(z)$ and its inverse are both Laurent polynomials; this is possible only in case $detP(z)$ is a monomial in z. Therefore $P(z)$ and $\tilde{P(z)}$ belong to GL(2;$\mathbf{R}[\mathbf{z}, \mathbf{z^{-1}}]$). Without loss of generality we can assume $detP(z) = 1$. Figure A.2 shows the polyphase representation of wavelet transform. The problem of finding an FIR wavelet transform

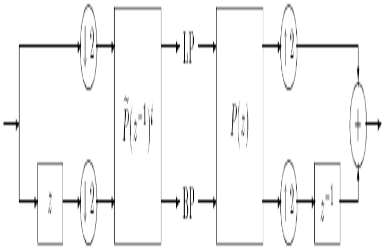

Figure A.2: Block Diagram for Polyphase representation for filter bank

thus amounts to finding a matrix $P(z)$ with determinant one. The most trivial example of a polyphase matrix is $P(z) = I$. This results in $h(z) = \tilde{h(z)} = 1$ and $g(z) = \tilde{g(z)} = 1$. This wavelet transform does nothing but subsampling even and odd samples and is often called as Lazy wavelet Transform. One can start from the Lazy wavelet and built wavelets with particular properties using the Lifting scheme which is explained below.

A.1.3 The Lifting Scheme

A filter pair (h, g) is said to be complimentary, if the determinant of the polyphase matrix is one. Let (h, g) be complementary. Then any other finite filter g^{new} complementary to h is of the form:

$$g^{new}(z) = g(z) + h(z)s(z^2) \tag{A.4}$$

where $s(z)$ is a Laurent Polynomial. Conversely any filter of this form is complementary to h. proof of this can be found in [**?**]. This is called a lifting of $g(z)$ using $h(z)$. After lifting, the new polyphase matrix is given by

$$P^{new}(z) = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$$

This operation doesnot change the determinant of the polyphase matrix. Figure A.3 shows the schematic representation of lifting. Similarly any filter h^{new} complementary to g is of the form

$$h^{new}(z) = h(z) - g(z)t(z^2) \tag{A.5}$$

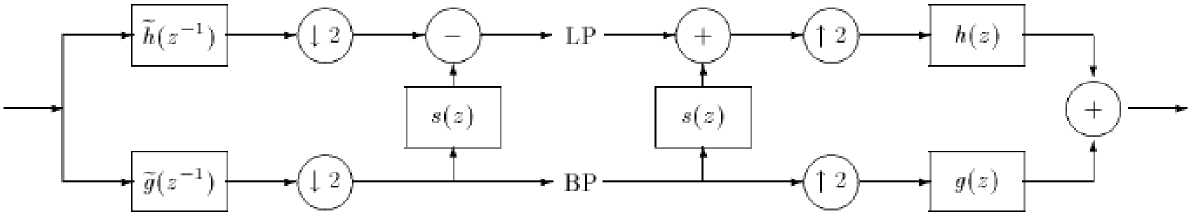

Figure A.3: Lifting Scheme.

where, $t(z)$ is a Laurent polynomial. This is called a dual lifting scheme. Wavelets can be built from the Lazy wavelet using these lifting and dual lifting steps.

A.1.4 Factorizing Algorithm

Any pair of complementary filters (h, g) can be factored into lifting steps using the Euclidean algorithm. Given a pair of complementary filters (h, g), then there always exist Laurent polynomials $s_i(z)$ and $t_i(z)$ for $1 \leq i \leq m$ and a non-zero constant K so that

$$P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$$

The dual polyphase matrix is given by

$$P(\tilde{z}) = \prod_{i=1}^{m} \begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix}$$

Figures A.3 and A.4 show the block diagrams for the lifting scheme of dual wavelets. Hence every finite wavelet filter can be obtained by starting with the Lazy wavelet followed by m lifting and dual lifting steps followed by a scaling. As shown in the above figures, the synthesis filter bank can be obtained by mirroring the analysis filter bank and changing signs of the filters. The PR is ensured by construction. This provides additional degrees of freedom in the design of filters, allowing any nonlinear into the basic blocks and any kind of signal extensions outside the borders. In particular, the integer wavelet transform is obtained by introducing rounding operator after each lifting scheme [**?**].

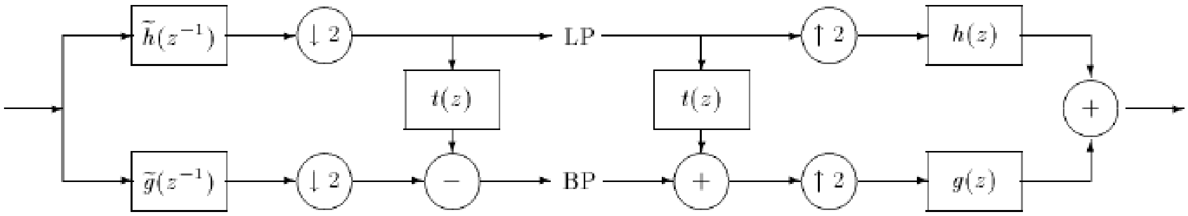

Figure A.4: Lifting Scheme.

Bibliography

[1] G. Menegaz and J. P. Thiran, "Lossy to lossless object-based coding of 3-d mri data," *IEEE Trans. Image Processing*, vol. 11, pp. 1053–1061, Sept. 2002.

[2] S. Wong, L. Zaremba, and H. Huang, "Radilogical image compression - a review," *Proc. IEEE*, vol. 83, pp. 194–219, Jan. 1995.

[3] P. Roos, M. Viergever, and M. V. Dijke, "Reversible intraframe compression of medical images," *IEEE Trans. Med. Imag.*, vol. 7, pp. 328–336, Dec. 1988.

[4] P. Roos and M. Viergever, "Reversible 3-d decorrelation of medical images," *IEEE Trans. Med. Imag.*, vol. 12, pp. 413–420, Sept. 1991.

[5] P. Roos and M. Viergever, "Reversible interframe compression of medical images : A comparison of decorrelation methods," *IEEE Trans. Med. Imag.*, vol. 10, pp. 538–547, Dec. 1991.

[6] H. Musmann, P. Pirsch, and H. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523–548, 1985.

[7] H. Lee, Y. Kim, A. H. Rowberg, and E. A. Riskin, "Statistical distributions of dct coefficients and their application to an interframe compression for 3-d medical images," *IEEE Trans. Med. Imag.*, vol. 12, pp. 478–485, Sept. 1993.

[8] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 339–355, June 1994.

[9] A. Nosratinia, N. Mohsenian, M. T. Orchard, and B. Liu, “Interframe coding of magnetic resonance images,” *IEEE Trans. Med. Imag.*, vol. 15, pp. 639–647, Oct. 1996.

[10] Y. Atulbasak and M. Tekalp, “Occlusion-adaptive, content-based mesh design and forward tracking,” *IEEE Trans. Image Processing*, vol. 6, pp. 1270–1281, Sept. 1997.

[11] J. Rissanen and G. Langdon, “Universal modeling and coding,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 12–27, Jan. 1981.

[12] T. V. Rambadran and K. Chen, “The use of contextual information in the reversible compression of medical images,” *IEEE Trans. Med. Imag.*, vol. 11, pp. 185–195, June 1992.

[13] M. Midtvik and I. Hovig, “Reversible compression of mr images,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 795–800, Sept. 1999.

[14] X. Wu and N. Memon, “Context-based, adaptive, lossless image coding,” *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

[15] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[16] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[17] D. Taubman and A. Zakhor, “Multirate 3-d subband coding of video,” *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

[18] D. Taubman, “High performance scalable image compression with ebcot,” *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[19] W. Sweldens, “The lifting scheme: A construction of second generation wavelet,” *SIAM J. Math. Anal.*, vol. 29, pp. 511–546, 1997.

[20] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.

[21] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-d spiht)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1374–1387, Dec. 2000.

[22] A. Munteanu, J. Cornelis, and P. Cristea, "Wavelet-based lossless compression of coronary angiographic images," *IEEE Trans. Med. Imag.*, vol. 18, pp. 272–281, Mar. 1999.

[23] H. Benoit-Cattin, A. Baskrut, F. Turjman, and R. Prost, "3d medical image coding using separable 3d wavelet decomposition and lattice vector quantization," *Signal processing, Elsevier*, vol. 59, pp. 139–153, 1997.

[24] J. Wang and H. Huang, "Medical image compression by using three-dimensional wavelet transform," *IEEE Trans. Med. Imag.*, Aug. 1996.

[25] Z. Xiong, X. Wu, S. Cheng, and J. Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Med. Imag.*, vol. 22, pp. 459–470, Mar. 2003.

[26] P. Schelkens, A. Munteanu, M. Galca, and X. Giro-Nieto, "Wavelet coding of volumetric medical datasets," *IEEE Trans. Med. Imag.*, vol. 22, pp. 441–458, Mar. 2003.

[27] C. Chrysafis and A. Ortega, "Efficient context based entropy coding for lossy wavelet image compression," in *Data Compression Conference*, 1997.

[28] X. Wu, "Context quantization with fisher discriminant for adaptive embedded wavelet image coding," in *Data Compression Conference*, 1999.

[29] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization and modeling," *IEEE Trans. Med. Imag.*, vol. 6, pp. 656–663, May 1997.

[30] Z. Liu and L. J. karam, "Context formation by mutual information maximization," in *Proc. Int. Conf. Image Processing (ICIP)*, IEEE, 2002.

[31] A. Tekalp, *Digital Video Processing.* Englewood Cliffs Newyork: Printice-Hall, 1995.

[32] Y. Wang and O. Lee, "Active mesh - a feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Processing*, vol. 3, pp. 610–624, Sept. 1994.

[33] B.K.P.Horn and B.G.Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[34] N. Jayant and P. Noll, *Digital Coding of Waveforms, Principles and Applications to speech and Video.* Englewood, New Jersey: Printice-Hall.

[35] S. Mallat, "A theory for multiresolution signal decomposition," *IEEE Trans. Pattern Ana. Machine Intell.*, vol. 11, pp. 674–693, July 1989.

[36] S. Maldonado-Bascon, F. Lopez-Ferreras, F. Acevedo-Rodriguez, and H. Gomez-Moreno, "Intra and inter-band information evaluation in still image coding by means of the wavelet transform," in *Proc. Int. Conf. Image Processing (ICIP)*, pp. 178–181, IEEE, 2000.

[37] M. Svensen, F. Kruggel, and D. Y. von Cramon, "Probabilistic modeling of single-trial fmri data," *IEEE Trans. Med. Imag.*, vol. 19, pp. 25–35, Jan. 2000.

[38] P. Bandettini, A. Jesmanowicz, E. Wong, and J. Hyde, "Processing strategies for time-course data sets in functional mri of the human brain," *Magn. Reson. Med*, vol. 30, pp. 161–173, 1993.

[39] Lee, "to be written,"

[40] E. Bullmore, M. Brammer, S. Williams, and S. R. Hesketh, "Statistical methods of estimation and inference for functional mr image analysis," *Magn. Reson. Med*, vol. 35, pp. 261–277, 1996.

[41] B. A. Ardekani, J. Kershaw, K. Kashikura, and I. Kanno, "Activation detection in functional mri using subspace modeling and maximum likelihood estimation," *IEEE Trans. Med. Imag.*, vol. 18, pp. 101–114, Feb. 1999.

[42] G. M. Boynton, S. A. Engel, G. H. Glover, and D. J. Heeger, "Linear systems analysis of functional magnetic resonance imaging in human v1," *J. Neurosci.*, vol. 16, pp. 4207–4221, July 1996.

[43] K. J. Friston, P. Jezzard, and R. Turner, "The analysis of functional mri time-series," *Human Brain Mapping*, vol. 1, pp. 153–171, 1994.

[44] K. J. Friston, A. Homes, J. Poline, P. Grasby, S. Williams, R. Frackowiak, and R. Turner, "The analysis of functional mri time-series revisited," *NeuroImage*, vol. 2, pp. 45–53, 1995.

[45] G. Marrelec, H. Benali, P. Ciuciu, and J. Poline, "Bayesian estimation of the hemodynamic response function in functional mri," in *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 21st International Workshop*, pp. 229–247, American Institute of Physics, 2002.

[46] G. Marrelec, H. Benali, P. Ciuciu, and J. Poline, "Robust bayesian estimation of the hemodynamic response function in event-related bold fmri using basic phsiological information," *Human Brain Mapping*, vol. 19, pp. 1–17, 2003.

[47] C. Goutte, F. A. Nielsen, and L. K. Hansen, "Modeling the haemodynamic response in fmri using smooth fir filters," *IEEE Trans. Med. Imag.*, vol. 19, pp. 1188–1201, Dec. 2000.

[48] M. A. Burock and A. M. Dale, "Estimation and detection of event-related fmri signals with temporally correlated noise: A statistically efficient and unbiased approach," *Human Brain Mapping*, vol. 11, pp. 249–260, 2000.

[49] F. Kruggel, D. von Cramen, and X. Descombes, "Comparison of filtering methods for fmri datasets," *NeuroImage*, vol. 10, pp. 530–542, 1999.

[50] R. Cox, "Improving the task-to-signal correlations in fmri by detrending slow components," in *Proc. ISMRM*, p. 1668, 1997.

[51] K. J. Friston, C. D. Frith, R. Turner, and R. S. J. Frackowiak, "Characterizing evoked hemodynamics with fmri," *NeuroImage*, vol. 2, pp. 157–165, 1995.

[52] G.-A. Hossein-Zadeh, B. A. Ardekani, and H. Soltanian-Zadeh, "Activation detection in fmri using a maximum energy ratio statistic obtained by adaptive spatial filtering," *IEEE Trans. Med. Imag.*, vol. 22, pp. 795–805, July 2003.

[53] P. L. Purdon and R. M. Weisskoff, "Effect of temporal autocorrelation due to physiological noise and stimulus paradigm on voxel-level false-positive rates in fmri," *Human Brain Mapping*, vol. 6, pp. 239–249, 1998.

[54] C. Gossl, D. P. Auer, and L. Fahrmeir, "Dynamic models in fmri,"

[55] F. G. Meyer, "Wavelet-based estimation of a semiparametric generalized linear model of fmri time-series," *IEEE Trans. Med. Imag.*, vol. 22, pp. 315–322, Mar. 2003.

[56] G. W. Wornell, "A karhunen-loeve-like expansion for $1/f$ processes via wavelets," *IEEE Trans. Inform. Theory*, vol. 36, pp. 859–861, Mar. 1998.

[57] J. L. Locascio, P. J. Jennings, C. I. Moore, and S. Corkin, "Time series analysis in the time domain and resampling methods for studies of functional magnetic brain imaging," *Human Brain Mapping*, vol. 5, pp. 168–193, 1997.

[58] K. J. Worsley and K. J. Friston, "Analysis of fmri time-series revisited-again," *NeuroImage*, vol. 2, pp. 173–181, 1995.

[59] E. Bullmore, C. Long, J. Suckling, J. Fadili, G. Calvert, F. Zelaya, T. Carpenter, and M. Brammer, "Colored noise and computational inference in neurophysiological (fmri) time series analysis: Resampling methods in time and wavelet domain," *Human Brain Mapping*, vol. 12, pp. 61–78, 2001.

[60] M. J. Fadili and E. T. Bullmore, "Wavelet-generalized least squares: A new blu estimator of linear regression models with $1/f$ errors," *NeuroImage*, vol. 15, pp. 217–232, 2002.

[61] P. Flandrin, “Wavelet analysis and synthesis of fractional brownian motion,” *IEEE Trans. Inform. Theory*, vol. 38, no. 12, pp. 910–917, 1992.

[62] G. W. Wornell, *Signal Processing with Fractals: A wavelet-based Approach.* Englewood, New Jersey: Printice-Hall, 1996.

[63] J. C. Rajapakse and J. Piyaratna, “Bayesian approach to segmentation of statistical parametric maps,” *IEEE Trans. Bioemed. Engg.*, vol. 48, pp. 1186–1194, Oct. 2001.

[64] X. Descombes, F. Kruggel, and D. Y. von Cramon, “Spatio-temporal fmri analysis using markov random fields,” *IEEE Trans. Med. Imag.*, vol. 17, pp. 1028–1039, Dec. 1998.

[65] C. A. B. Sea Chen and M. J. Lowe, “Clustered components analysis for functional mri,” *IEEE Trans. Med. Imag.*, vol. 23, pp. 85–98, Jan. 2004.

[66] M. W. Woolrich, M. Jenkinson, J. M. Brady, and S. M. Smith, “Fully bayesian spatio-temporal modeling of fmri data,” *IEEE Trans. Med. Imag.*, vol. 23, pp. 213–231, Feb. 2004.

[67] M. E. Tipping and C. M. Bishop, “Probabilistic principal component analysis,” in *Neural Computing Research Group (NCRG), Technical Report NCRG/97/010.*

[68] P. Gowland, S. Francis, P. Morris, and R. Bowtell, “Watching the brain at work,” *Physics World*, pp. 31–35, Dec. 2002.

[69] S. Kay, *Fundamentals of Statistical Signal Processing : Estimation Theory.* New Jersey: Printice-Hall, 1995.

[70] S. Geman and D. Geman, “Stochastic relaxation, gibbs distribution, and the bayesian restoration of images,” *IEEE Trans. Pattern Ana. Machine Intell.*, vol. 6, pp. 721–741, June 1984.

[71] Y. Zhang, M. Brady, and S. Smith, “Segmentation of mr images through a hidden markov random field and the expectation-maximization algorithm,” *IEEE Trans. Med. Imag.*, vol. 20, pp. 721–741, Jan. 2001.

[72] L.R.Frank, R.B.Buxton, and E.C.Wong, “Probabilistic analysis of functional magnetic resonance imaging data,” *Magn. Reson. Med*, vol. 39, pp. 132–148, 1998.

[73] E. Zarahn, G. K. Aguirre, and M. D’Esposito, “Empirical analyses of bold fmri statistics i. spatially unsmoothed data collected under null hypothesis conditions,” *NeuroImage*, vol. 5, pp. 179–197, 1997.

[74] A. H. Tewwfik and M. Kim, “Correlation structure of the discrete wavelet coefficients of fractional brownian motion,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 904–909, 1992.

[75] R. W. Dijkerman and R. R. Mazumdar, “On the correlation structure of the wavelet coefficients of fractional brownian motion,” *IEEE Trans. Inform. Theory*, vol. 40, pp. 1609–1612, 1994.

[76] B. Ninness, “Estimation of $1/f$ noise,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 32–46, 1998.

[77] D. P. Percival, “On estimation of wavelet variance,” *Biometrika*, vol. 82, pp. 619–631, 1995.