Weierstraß-Institut
für Angewandte Analysis und Stochastik

im Forschungsverbund Berlin e.V.

Preprint ISSN 0946 – 8633

Structural Adaptive Smoothing in Diffusion Tensor Imaging: the R package dti

Jörg Polzehl[1], Karsten Tabelow [1 2]

submitted: 5th December 2008

[1] Weierstrass Institute
for Applied Analysis and Stochastics,
Mohrenstr. 39, 10117
Berlin, Germany
E-Mail: tabelow@wias-berlin.de

No. 1382

Berlin 2008

[2]Supported by the DFG Research Center MATHEON "Mathematics for key technologies" in Berlin

2000 *Mathematics Subject Classification.* 62P10, 92C55, 62G05 .

Key words and phrases. Structural Adaptive Smoothing, Diffusion Weighted Imaging, Diffusion Tensor Model, Rician Bias, Software.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Mohrenstraße 39
10117 Berlin
Germany

Fax: + 49 30 2044975
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract

Diffusion Weighted Imaging has become and will certainly continue to be an important tool in medical research and diagnostics. Data obtained with Diffusion Weighted Imaging are characterized by a high noise level. Thus, estimation of quantities like anisotropy indices or the main diffusion direction may be significantly compromised by noise in clinical or neuroscience applications.

Here, we present a new package **dti** for R, which provides functions for the analysis of diffusion weighted data within the diffusion tensor model. This includes smoothing by a recently proposed structural adaptive smoothing procedure based on the Propagation-Separation approach in the context of the widely used Diffusion Tensor Model. We extend the procedure and show, how a correction for Rician bias can be incorporated. We use a heteroscedastic nonlinear regression model to estimate the diffusion tensor. The smoothing procedure naturally adapts to different structures of different size and thus avoids oversmoothing edges and fine structures.

We illustrate the usage and capabilities of the package through some examples.

1 Introduction

The basic principles of magnetic resonance Diffusion Weighted Imaging (DWI) were introduced in the 1980's [1, 2, 3], after nuclear magnetic resonance (NMR) had long been known to be sensitive to diffusion of molecules in complex systems [4]. Since then, DWI has evolved into a versatile tool for in-vivo examination of tissues in the human brain [5] and spinal cord [6]. DWI probes microscopic structures well beyond typical image resolutions through water molecule displacement, a fact attracting broad interest in this technique. It can be used in particular to characterize the integrity of neuronal tissue in the central nervous system.

NMR images are acquired from signals measured in k-space by fast Fourier transform. In the case of single receiver coil systems and additive complex Gaussian noise in k-space this results in image grey values that follow a Rician distribution $Rice(\theta, \sigma)$ [7]. This distribution is characterized by two parameters: θ, corresponding to the signal of interest, and σ related to the noise standard deviation in k-space.

Diffusion in neuronal tissue depends on the particular microscopic structure of the tissue

and is usually not isotropic. Different diffusion directions $\vec{b}$ can be probed by applying bipolar magnetic field diffusion gradients [8]. Let $S_{\vec{b}}$ denote the acquired diffusion weighted image with the "b-value" b, which depends on the pulse sequence parameters. The non-diffusion weighted image ($b = 0$) is denoted by S_0. Due to the diffusion of water molecules during the application of the magnetic field gradient, the signal $S_{\vec{b}}$ is attenuated relative to S_0:

$$S_{\vec{b}} \sim Rice(\theta_0 \exp(-bD(\vec{b})),\ \sigma)\ , \quad S_0 \sim Rice(\theta_0,\ \sigma) \tag{1}$$

where $D(\vec{b})$ is the diffusion constant probed in direction $\vec{b}$. Data are usually measured for $N_{grad} > 16$ different gradient directions, including $b = 0$.

Equation (1) imposes several problems for the data analysis. I. The model for the measured data S is nonlinear in D. II. The measured data S suffers from noise typical for NMR images. While for the non-diffusion weighted image S_0 a Gaussian noise model seems to be an appropriate approximation, the Rician distribution of the attenuated signal values $S_{\vec{b}}$ can be significantly different from a Gaussian. Ignoring this introduces a bias into the estimation of the diffusion constant and hence all derived quantities. III. The lower the noise level in the measured data is, the more accurately the diffusion constant can be estimated. Thus, a noise reduction would be helpful to improve the accuracy of the diagnostic measures derived from the data.

Typically, diffusion weighted images $S_{\vec{b}}$ are measured for 15 to 100 diffusion gradient directions $\vec{b}$ resulting in very high-dimensional data. It was the development of Diffusion Tensor Imaging (DTI) [9, 10] which triggered a plethora of clinical and neuroscience applications of DWI. There, the information in the diffusion weighted images is reduced to a three dimensional Gaussian distribution model for free anisotropic diffusion. Within this model, diffusion is completely characterized by a rank-2 diffusion tensor $\mathcal{D}$, a symmetric positive definite 3×3 matrix with six independent components. This model describes diffusion completely if the microscopic diffusion properties within a voxel are homogeneous and non-restricted. In the presence of partial volume effects, like crossing fibers, this model is only an approximation. For these cases, more sophisticated models exist, which rely on the measurement of diffusion weighted images with high angular resolution [11, 12]. These shall not be considered in this paper.

DTI suffers from significant noise which may render subsequent analysis or medical decisions more difficult. In the presence of noise the estimation of anisotropy indices from the diffusion tensor is known to be systematically biased [13, 14]. In addition to the common random errors the order of the diffusion eigenvectors of the tensor which is essential for fiber tracking is subject to a sorting bias especially at high noise levels. Noise reduction is therefore essential.

In [15] we proposed a new smoothing method for noise reduction in DTI based on the Propagation-Separation (PS) approach [16]. The procedure has been shown to naturally adapt to the structures of interest at different scales [15, 16]. Thus, it avoids loss of information on size and shape of structures, which is typically observed when using non-adaptive filters. This is especially important for DTI, where the structures of interest, the white matter fibers, may be very small and anisotropic.

Extending [15] we will in this paper discuss the estimation of the diffusion tensor from the diffusion weighted data using heteroscedastic non-linear regression, propose a model for heteroscedastic variances and provide a solution for Rician bias correction within the structural adaptive smoothing procedure. We present a new package **dti** for R, which implements this extended PS approach for reducing the noise by adaptive smoothing of diffusion weighted data in the context of the diffusion tensor model. The package provides methods for all steps in a common diffusion tensor analysis from data access to visualization of estimated tensors and indices.

The paper is organized as follows: In section 2 we review the basic notation of DTI and discuss linear as well as non-linear estimation methods for the diffusion tensor and the estimation of heteroscedastic variances. Section 3 is dedicated to Rician bias and its correction. We outline the extended structural adaptive smoothing procedure in section 4. Finally, we describe the usage and capabilities of the R-package **dti** and provide several examples in the last section 5.

2 Diffusion Tensor Imaging

2.1 The diffusion tensor model and its quantities

Using a Gaussian model of diffusion, the anisotropy can be described by a rank-2 diffusion tensor $\mathcal{D}$, which is represented by a symmetric positive definite 3×3 matrix:

$$\mathcal{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} . \tag{2}$$

The "b-value" in Eq.(1) is replaced by a "b-matrix", [17, 18], leading to

$$S_{\vec{b}} \sim Rice(\theta_0 \exp(-\vec{\mathbf{b}}\vec{\mathcal{D}}),\ \sigma) \tag{3}$$

where $(b_{ij})_{i,j=x,y,z}$ denotes the matrix $\mathbf{b} = b \cdot \vec{b}\ \vec{b}^{\top}$, $\vec{\mathbf{b}} = (b_{xx}, b_{yy}, b_{zz}, 2b_{xy}, 2b_{xz}, 2b_{yz})$ and $\vec{\mathcal{D}} = (D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{xz}, D_{yz})^{\top}$.

The components of the diffusion tensor clearly depend on the orientation of the object in the scanner frame xyz. Only rotationally invariant quantities derived from the diffusion tensor circumvent this dependence and are usually used for further analysis, mainly based on the eigenvalues μ_i $(i = 1, 2, 3)$ of $\mathcal{D}$ with $\mu_i > 0$ for positive definite tensors. The eigenvector $\vec{e}_1$ corresponding to the principal eigenvalue μ_1 determines the main diffusion direction used for fiber tracking.

The simplest quantity based on the eigenvalues is the trace of the diffusion tensor

$$Tr(\mathcal{D}) = \sum_{i=1}^{3} \mu_i \ . \tag{4}$$

which is related to the mean diffusivity $\langle\mu\rangle = Tr(D)/3$. The anisotropy of the diffusion can be described using higher moments of the eigenvalues μ_i. The widely used fractional anisotropy (FA) is defined as

$$FA = \sqrt{\frac{3}{2}} \sqrt{\sum_{i=1}^{3} (\mu_i - \langle\mu\rangle)^2 \ / \ \sum_{i=1}^{3} \mu_i^2} \tag{5}$$

with $0 \leq FA \leq 1$, where $FA = 0$ indicates equal eigenvalues and hence no diffusion anisotropy.

The resulting FA-maps together with a color-coding scheme are helpful for medical diagnostics. The principal eigenvector $\vec{e}_1 = (e_{1x}, e_{1y}, e_{1z})$ is used for assigning each voxel a specific color, interpreting e_{1x}, e_{1y}, and e_{1z} as red, green, and blue contribution weighted with the value FA:

$$(R, G, B) = (|e_{1x}|, |e_{1y}|, |e_{1z}|) \cdot FA \tag{6}$$

In contrast to the eigenvalues and the fractional anisotropy the principal eigenvector depends on the orientation of the object in the scanner frame xyz. Thus, the color coding depends on patient orientation.

The multiplication with the FA-value in Eq.(6) leads to darker areas with low fractional anisotropy in contrast to bright white matter areas with large values for FA. However, the intensity of a color image is approximately given by $R+G+B = (|e_{1x}|+|e_{1y}|+|e_{1z}|) \cdot FA$ which depends on the orientation of the object. Additionally, a grey value reproduction of such a color-coded FA-map may show similar grey values for different FA-values and vice versa. One could therefore use a slightly different color encoding

$$(R, G, B) = (e_{1x}^2, e_{1y}^2, e_{1z}^2) \cdot FA \ ,$$

since due to eigenvector normalization we have $R+G+B = (e_{1x}^2 + e_{1y}^2 + e_{1z}^2) \cdot FA = FA$. Also note, that the direction corresponding to a specific color is not unique since the

signs of the eigenvector components are lost. This problem can be circumvented when the diffusion tensor ellipsoid is shown in 3D using the color scheme.

As an alternative to the fractional anisotropy the geodesic anisotropy (GA)

$$GA = \sqrt{\sum_{i=1}^{3}(log(\mu_i) - \overline{log(\mu)})^2}, \quad \overline{log(\mu)} = \frac{1}{3}\sum_{i=1}^{3} log(\mu_i) \tag{7}$$

has been proposed [19, 20] to take the metric structure of the tensor space into account. Note, that $0 \leq GA < \infty$.

The diffusion tensor represents a diffusion ellipsoid with the three main axis given by μ_i. To describe the shape of the diffusion tensor a decomposition into three basic shapes – spherical, planar, and linear – can be used [21, 22]:

$$\begin{aligned} C_s &= \frac{\mu_3}{\langle\mu\rangle} \\ C_p &= \frac{2(\mu_2 - \mu_3)}{3\langle\mu\rangle} \\ C_l &= \frac{(\mu_1 - \mu_2)}{3\langle\mu\rangle}\ , \end{aligned} \tag{8}$$

where the sum of the three contributions is $C_s + C_p + C_l = 1$. Again, we can interpret the three shape values as red, green, and blue component of a color assigned to the voxel in order to visualize the tensor shape.

2.2 Linear estimation of the diffusion tensor

To completely determine the diffusion tensor, one has to acquire diffusion weighted images for at least $N_{grad} = 7$ gradient directions $\vec{b}$ including the non-diffusion-weighted image S_0. However, since the estimation errors of the eigenvalues are not rotationally invariant, about 30 gradient directions are required for a robust quantitative measurement of eigenvalues and tensor orientation [23, 24, 25].

In one of our previous publications [15] at each voxel i the diffusion tensor $\mathcal{D}_i$ has been estimated via multiple linear regression using the equation:

$$-\ln \frac{S_{\vec{b},i}}{S_{0,i}} = \vec{\mathbf{b}}\vec{\mathcal{D}}_i + \varepsilon_{\vec{b},i}\ , \tag{9}$$

and the assumption that the errors $\varepsilon_{\vec{b},i}$ are i.i.d. Gaussian $\mathcal{N}(0, \sigma_i)$. The noise variance σ_i^2 in voxel i has been estimated from the residuals in the linear model, while the variance of the estimated tensor was assessed from this estimate and the "b-matrix". There are at least two weaknesses in this approach. First, due to the *log*-transform, error variances

are highly heteroscedastic and therefore the ordinary least squares estimate is inefficient and its variance estimate biased. Even more important, if the tensor model is inadequate the residuals from (9) contain structure that is not explained by the model and hence, the variance of the estimated tensor may be strongly over-estimated. This decreases the statistical penalty in the adaptive smoothing procedure proposed in [15] and therefore reduces the effectivity of adaptation.

2.3 Non-linear estimation of the diffusion tensor

The drawbacks connected with the use of the *log*-transform of the data in the linearized model (9) can be avoided by using the heteroscedastic non-linear regression model

$$S_{\vec{b},i} \quad = \quad \theta_{0,i} \exp(-\vec{\mathbf{b}}\vec{\mathcal{D}}_i) + \varepsilon_{\vec{b},i} \ , \quad \mathbf{E}\varepsilon_{\vec{b},i} = 0 \ , \quad \mathbf{Var}\varepsilon_{\vec{b},i} = \sigma^2_{\vec{b},i} \ . \tag{10}$$

for all $\vec{b}$ including the non-diffusion weighted gradient ($b = 0$). In this model the parameter θ_0 reflects the expected intensity in non-diffusion weighted images. In order to enforce positive definiteness of the tensor estimates it is possible to re-parametrize the model (10) using the Choleski decomposition $\mathcal{D} = \mathcal{R}^{\mathcal{T}}\mathcal{R}$ with an upper triangular matrix $\mathcal{R}$ with non-zero diagonal elements [26]. In [27] the efficiency of this approach in comparison with an unconstrained nonlinear regression model (10) has been investigated. In our implementation we estimate the parameters $\theta_{0,i}$ and $\mathcal{D}_i$ in each voxel i using the risk function $\mathbf{R}$ defined by

$$\mathbf{R}(S_{\cdot,i}, \theta, \mathcal{D}) = \sum_{\vec{b}} \frac{(S_{\vec{b},i} - \theta \exp(-\vec{\mathbf{b}}\vec{\mathcal{D}}))^2}{\sigma^2_{\vec{b},i}} \tag{11}$$

where the sum is over all diffusion gradient vectors $\vec{b}$ including the non-diffusion weighted images ($b = 0$). In

$$\begin{pmatrix} \widehat{\theta}_{0,i} \\ \widehat{\mathcal{D}}_i \end{pmatrix} = \arg\min_{\theta,\mathcal{D}} \mathbf{R}(S_{\cdot,i}, \theta, \mathcal{D}) \tag{12}$$

we use an unconstrained minimization as long as the estimated diffusion tensor is positive definite and the re-parametrization $\mathcal{D} = \mathcal{R}^{\mathcal{T}}\mathcal{R}$ [26], otherwise. Minimization is done using a regularized Gauss-Newton algorithm [28]. The variability of the tensor estimates can be assessed using profile likelihood.

2.4 Variance estimates

In order to estimate the diffusion tensor using (11) and (12) we need to model the heteroscedastic variances $\sigma^2_{\vec{b},i}$. Empirical evidence from various DWI data sets with replicated

non-diffusion weighted images suggests the existence of a wide range of image intensity values where the standard deviation $\sigma_{\vec{b},i}$ can be well approximated by a linear function of the observed image intensity, see Figure 1 for two examples.

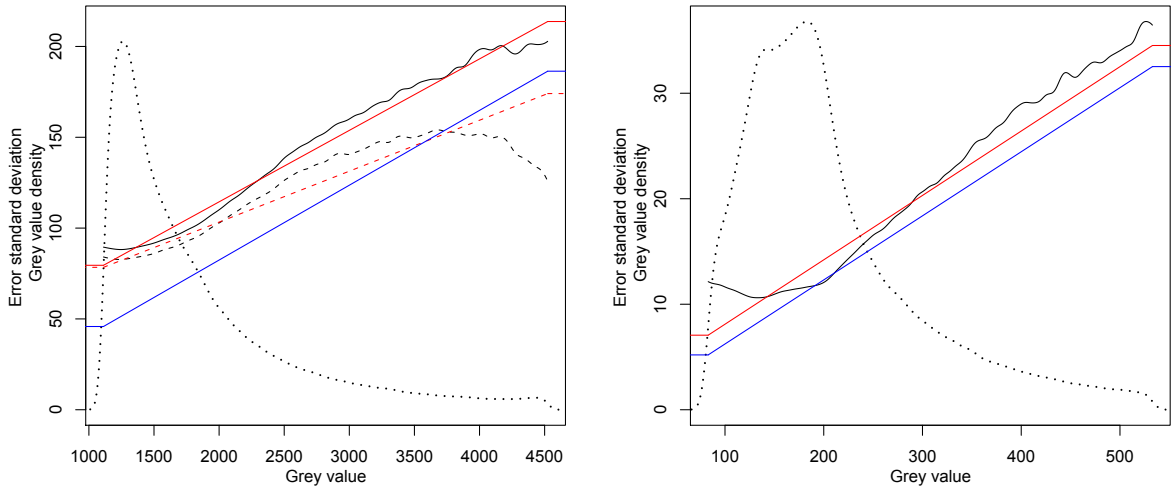

Figure 1: Local polynomial estimates of mean standard deviation as a function of mean grey value (solid) and density of mean grey values (dotted) observed for two DWI data sets with replicated non-diffusion weighted images S_0. The red and blue curves correspond to variance estimates obtained from replicates (red) and mean variance estimates obtained from single images (blue), respectively, using the model (13) in both cases. The left data set is not registered causing a positive bias in voxelwise variance estimates from replications. The dashed curves in the left panel provide the corresponding estimates using mean absolute deviation (mad) as an alternative to voxelwise standard deviation (sd) to lower this effect.

We therefore use the following model for the error standard deviations

$$\sigma_{\vec{b},i} = \begin{cases} \sigma_0 + \sigma_1 A_0 & \theta_{\vec{b},i} < A_0 \\ \sigma_0 + \sigma_1 \theta_{b,i} & A_0 \leq \theta_{\vec{b},i} < A_l \\ \sigma_0 + \sigma_1 A_1 & A_1 \leq \theta_{\vec{b},i} \end{cases} \tag{13}$$

Here A_0 is set to the minimum, over voxel within the head, intensity in non-diffusion weighted images. For A_1 we use the 0.99 quantile of S_0 intensities within the head. The choice of A_0 and A_1 coincides with the range where we observe approximative linearity between the standard deviation and the mean.

The parameters σ_0 and σ_1 are estimated by linear regression between estimated voxelwise

standard deviations and mean grey values in the case of a replicated non-diffusion weighted image or from a single non-diffusion weighted image using adaptive smoothing with explicit specification of the dependency between mean and standard deviation, see [29]. In both cases the estimates will be restricted to use voxel with intensity within the range (A_0, A_1).

The linear dependence between noise variance and image intensity seems to reflect properties of physiological noise.

3 Handling Rician bias in diffusion weighted images

3.1 Rician bias

Termal noise in DWI can be modeled as additive Gaussian noise in both the real and imaginary part of the signal in k-space. After fast Fourier transform into image space the resulting observed signal follows a Rician distribution [7] with parameters ζ and σ. ζ is the signal of interest while σ corresponds to the standard deviation of the errors in k-space. The density of Rician distribution is given by

$$p(x) = \frac{x}{\sigma^2} \exp(-\frac{x^2+\zeta^2}{2\sigma^2}) I_0(\frac{x\zeta}{\sigma^2}), \tag{14}$$

where I_0 is the modified zeroth-order Bessel function of the first kind. Mean and variance of the Rician distribution are given by

$$\mathbf{E}X = \sigma\sqrt{\pi/2} L_{1/2}(-\zeta^2/2\sigma^2) \tag{15}$$

$$\mathbf{D}X = 2\sigma^2 + \zeta^2 - \frac{\pi\sigma^2}{2} L^2_{1/2}(-\zeta^2/2\sigma^2) \tag{16}$$

with

$$L_{1/2}(x) = e^{x/2}[(1-x)I_0(-x/2) - xI_1(-x/2)]. \tag{17}$$

For large ζ/σ we get $\mathbf{E}X \approx \zeta$, while for small ζ/σ the expected value of the observed signal is significantly larger than the parameter of interest ζ. This effect is called Rician bias and is more pronounced in the diffusion weighted images where the signal is attenuated, see Eq.(1). The Rician bias in the diffusion weighted images may lead to a bias in the estimated tensors as well as in quantities derived from the tensor, see [30]. We therefore include a correction for Rician bias in our implementation.

3.2 Correction for Rician bias

In order to avoid the Rician bias we need to estimate the parameters ζ and σ of the underlying Rician distribution from the measured signals S. Let us assume we have

samples $\mathcal{S}_n = \{S_{1,k}, \ldots, S_{n,k}\}_{k=1}^{N_{grad}}$ drawn from a Rician distribution $Rice(\zeta_k, \sigma)$. This resembles the situation within DWI data assuming that the noise variance in k-space does not depend on the gradient direction. For identifiability of the distribution parameters we need $n > 1$, which can be achieved by locating voxel with similar parameters within a local vicinity, see section 4. Let now $W = \{w_1, \ldots, w_n\}$ define a set of weights. We can then define a weighted log-likelihood function as

$$l(\mathcal{S}_n; \zeta, \sigma, W) = \sum_{k=1}^{N_{grad}} \sum_{j=1}^{n} w_j \left(\log \frac{S_{j,k}}{\sigma^2} - \frac{S_{j,k}^2 + \zeta_k^2}{2\sigma^2} + \log I_0 \left(\frac{S_{j,k}\zeta_k}{\sigma^2} \right) \right). \tag{18}$$

Differentiating with respect to the parameters ζ and σ^2 yields conditions for the likelihood estimate of $\zeta = (\zeta_1, \ldots, \zeta_{N_{grad}})$ and σ^2

$$\begin{aligned} \frac{d}{d\zeta_k} l(\mathcal{S}_n; \zeta, \sigma, W) &= \sum_{j=1}^{n} w_j \frac{I_1\left(\frac{S_{j,k}\zeta_k}{\sigma^2}\right)}{I_0\left(\frac{S_{j,k}\zeta_k}{\sigma^2}\right)} S_{j,k} - \zeta_k \sum_{j=1}^{n} w_j = 0 \\ \frac{d}{d\sigma^2} l(\mathcal{S}_n; \zeta, \sigma, W) &= \sum_{k=1}^{N_{grad}} \sum_{j=1}^{n} w_j \left(-\frac{1}{\sigma^2} + \frac{S_{j,k}^2 + \zeta_k^2}{2\sigma^4} - \frac{I_1\left(\frac{S_{j,k}\zeta_k}{\sigma^2}\right)}{I_0\left(\frac{S_{j,k}\zeta_k}{\sigma^2}\right)} \frac{S_{j,k}\zeta_k}{\sigma^4} \right) = 0. \end{aligned} \tag{19}$$

The estimates $\widehat{\zeta_k}$ and $\widehat{\sigma^2}$ can thus be obtained as fixpoints of

$$\begin{aligned} \widehat{\zeta_k} &= \frac{1}{\sum_{j=1}^{n} w_j} \sum_{j=1}^{n} w_j \frac{I_1\left(\frac{S_{j,k}\widehat{\zeta_k}}{\widehat{\sigma^2}}\right)}{I_0\left(\frac{S_{j,k}\widehat{\zeta_k}}{\widehat{\sigma^2}}\right)} S_{j,k} \\ \widehat{\sigma^2} &= \frac{1}{N_{grad} \sum_{j=1}^{n} w_j} \sum_{k=1}^{N_{grad}} \sum_{j=1}^{n} w_j \left(\frac{S_{j,k}^2 + \widehat{\zeta_k}^2}{2} - \frac{I_1\left(\frac{S_{j,k}\widehat{\zeta_k}}{\widehat{\sigma^2}}\right)}{I_0\left(\frac{S_{j,k}\widehat{\zeta_k}}{\widehat{\sigma^2}}\right)} S_{j,k}\widehat{\zeta_k} \right) \end{aligned} \tag{20}$$

through iteration. As initial estimate we use the corresponding likelihood estimates for a Gaussian distribution

$$\begin{aligned} \widehat{\zeta_k}^{(0)} &= \frac{1}{\sum_{j=1}^{n} w_j} \sum_{j=1}^{n} w_j S_{j,k} \\ \widehat{\sigma^2}^{(0)} &= \frac{(\sum_{j=1}^{n} w_j)^2}{(\sum_{j=1}^{n} w_j)^2 - \sum_{j=1}^{n} w_j^2} \frac{1}{N_{grad} \sum_{j=1}^{n} w_j} \sum_{k=1}^{N_{grad}} \sum_{j=1}^{n} w_j (S_{j,k} - \widehat{\zeta_k})^2. \end{aligned} \tag{21}$$

We use a prespecified number of iteration steps depending on the ratio $\widehat{\zeta}_k^{(0)}/\widehat{\sigma}_k^{(0)}$, i.e. no iteration if the ratio is larger than 10 and up to 6 iterations if the ratio is small.

4 Structural Adaptive Smoothing of DWI data within the diffusion tensor model

We recently proposed a new structural adaptive smoothing algorithm for diffusion weighted data in the context of the diffusion tensor model [15]. The approach reduces the error of the estimated tensor directions and tensor characteristics like fractional anisotropy by smoothing the observed DWI data. The application of a standard Gaussian filter would be highly inefficient in the DTI applications in view of the anisotropic nature of the diffusion tensor and sharp boundaries between region with different tensor characteristics. Indeed, the tensor direction remains constant mainly along the fiber directions. Averaging over a large symmetric neighborhood of every voxel would thus lead to a loss of directional information.

In order to avoid such a loss our smoothing procedure sequentially determines at increasing scales local weighting schemes with positive weights for voxel that show similar characteristics. To achieve this we employ the structural assumption that for every voxel there exists a vicinity in which the diffusion tensor is nearly constant. This assumption reflects the fact that the structures of interest are regions with a homogeneous fractional anisotropy, a homogeneous diffusivity, and a locally constant direction field. The shape of this neighborhood can be quite different for different voxel and cannot be described by few simple characteristics like bandwidth or principal directions.

The algorithm for the case of linear tensor estimates using model (9) has been described in detail in [15]. Here, we shortly present a modified algorithm that is based on the *nonlinear regression model* (10) for the diffusion tensor and incorporates the *Rician bias correction* developed in the previous section:

- **Initialization:** Set $k = 1$, initialize the bandwidth $h^{(1)} = c_h$. For each voxel i initialize $\widehat{\zeta}^{(0)}_{\vec{b},i} = S_{\vec{b},i}$ with the data, and $\widehat{\mathcal{D}}^{(0)}_i$ and $\widehat{\theta}^{(0)}_{0,i}$ by equation (12), set $N^{(0)}_i = 1$. Estimate the parameters σ_0 and σ_1 of the variance model (13).

- **Adaptation:** For each voxel pair i, j, we compute the penalty

$$s^{(k)}_{ij} \;=\; \frac{N^{(k-1)}_i}{\lambda C_i(g, h^{(k-1)})} \left[\mathbf{R}\left(\widehat{\zeta}^{(k-1)}_{\cdot,i}, \widehat{\theta}^{(k-1)}_{0,j}, \widehat{\mathcal{D}}^{(k-1)}_j\right) - \mathbf{R}\left(\widehat{\zeta}^{(k-1)}_{\cdot,i}, \widehat{\theta}^{(k-1)}_{0,i}, \widehat{\mathcal{D}}^{(k-1)}_i\right)\right]$$

  with the risk $\mathbf{R}$ based on the previous estimates $\widehat{\zeta}^{(k-1)}_{\cdot,i}$ at voxel i. $s^{(k)}_{ij}$ measures the statistical difference between the estimates $\widehat{\theta}^{(k-1)}_0$ and $\widehat{\mathcal{D}}^{(k-1)}$ at voxel i and j. Weights are computed as

$$w^{(k)}_{ij} = K_{\mathrm{loc}}\left(\Delta(i, j, \widetilde{\mathcal{D}}^{(k-1)}_i)/h^{(k)}\right) K_{\mathrm{st}}\left(s^{(k)}_{ij}\right) \;,$$

with appropriate kernel functions K_{loc} and K_{st}, an anisotropic distance function Δ, and a regularized tensor estimate $\widetilde{\mathcal{D}}_i^{(k-1)}$, see [15] for details.

- **Rice bias correction and estimation of diffusion weighted images:** Compute $\widehat{\zeta}_{.,i}^{(k)} = (\widehat{\zeta}_{1,i}^{(k)}, \ldots \widehat{\zeta}_{N_{grad},i}^{(k)})$ by maximizing the log-likelihood (18)

$$\widehat{\zeta}_{.,i}^{(k)} = \arg_{\zeta} \max_{\zeta,\sigma} \; l(\mathcal{S}_n; \zeta, \sigma, W_i^{(k)})$$

  using the weighting scheme $W_i^{(k)} = (w_{i1}^{(k)}, \ldots, w_{in}^{(k)})$ and evaluating equations (21,20).

- **Parameter estimation:** Compute new estimates of the expected non-diffusion weighted images $\theta_{0,i}$ and diffusion tensors $\mathcal{D}_i$ as

$$\begin{pmatrix} \widehat{\theta}_{0,i}^{(k)} \\ \widehat{\mathcal{D}}_i^{(k)} \end{pmatrix} = \arg\min_{\theta,\mathcal{D}} \mathbf{R}(\widehat{\zeta}_{.,i}^{(k)}, \theta, \mathcal{D}),$$

  see (12). Set $N_i^{(k)} = \sum_{j=1}^{n} w_{ij}^{(k)}$.

- **Stopping:** Stop if $k = k^*$ for a preselected number of iteration steps, otherwise set $h^{(k+1)} = c_h h^{(k)}$, increase k by 1 and continue with the adaptation step.

The term $C_i(g, h^{(k-1)})$ provides an adjustment under the assumption that spatial smoothness of the errors can be modeled by a convolution of independent errors with a Gaussian kernel of bandwidth g, see e.g. [15] or [31] for details. The Rician bias correction can be omitted using $S_{.,i}$ instead of $\widehat{\zeta}_{.,i}^{(k)}$ in all steps. λ is the main parameter of the procedure and can be determined by simulations, see [16, 15] for details.

5 Using the package dti

This document refers to the version 0.6-0 of the **dti**-package which is available from CRAN `http://cran.r-project.org`. The software is under constant development, see subsection 5.8 for details on the plans for the next future. Changes are documented in the HISTORY file of the package.

For the analysis of diffusion weighted data, there is an overlap in functionality needed from other packages. In order to fully use the package **dti** it is therefore required to install the packages **fmri** for reading and writing medical imaging formats like ANALYZE, NIfTI, or DICOM as well as **adimpro** and **rgl** for visualization. These packages can also be downloaded from CRAN.

For a typical analysis we assume the DWI data to reside in a directory, say ‘‘datadir’’, and the gradient matrix in a file, say ‘‘gradient.txt‘‘. A script for the analysis could then have the form

```
R> grad <- read.table(''gradient.txt‘‘)

R> data <- readDWIdata(grad,''datadir‘‘,dataformat,ngrad)

R> data <- sdpar(data)

R> tensor <- dtiTensor(data)

R> tensor <- dti.smooth(data,hmax=4)

R> dtind <- dtiIndices(tensor)
```

The steps of the analysis including visualization and some utility functions will be discussed within the next subsections. Memory requirements and computation time will also be illustrated.

For an overview over the package capabilities see `help(dti)` and run the `dti_art` demo

```
R> demo(dti_art)
```

It creates the artificial dti dataset used in [15].

5.1 Data processing

Diffusion weighted data can often be found collected in (one or several) DICOM folders depending on the acquisition protocol of the scanner, as well as preprocessed compilations of ANALYZE or NIfTI files or other formats. The package provides a function `readDWIdata()` to read data in these formats from one or more directories that contain solely the imaging files. e.g. DICOM files for all slices and gradient directions. There may be cases, where the slice ordering differs from the alphabetic order of the files. In these cases the `order` argument of the function can be used.

A region of interest can be specified by index vectors `xind`, `yind`, `zind` for the three dimensions of the data cubes. Reading a region-of-interest (ROI) of DICOM data from two directories is e.g. performed by

```
R> grad <- read.table(''gradient.txt‘‘)

R> dwiobj <- readDWIdata(grad, c("datadir/s0011/", "datadir/s0012/"),

            "DICOM",72,xind=129:196,yind=129:196,zind=25:30).
```

The two-dimensional array `grad` has rows corresponding to the gradient vectors including

the non-diffusion weighted gradient $(0, 0, 0)$ in the same order as the diffusion weighted data.

As an alternative we provide a simple interface for the diffusion weighted data using a binary file. Such a file can be created using the I/O-funtions from the packages **fmri**, **AnalyzeFMRI**, **DICOM** or others. Diffusion weighted data is a set of N_{grad} three dimensional datasets (e.g. covering the brain) corresponding to N_{grad} diffusion weighted images including the non-diffusion weighted images. These datasets are written in the corresponding order into a binary file of 2 byte integer values, denoted by `S-all` here:

```
R> con <- file("S-all","wb")
R> for (gg in 1:16) {
R>    data <- read.ANALYZE(filename[gg])
R>    writeBin(as.integer(extract.data(data)),con,2)
R> }
R> close(con)
```

where `filename` refers to the array of file names for the ANALYZE files containing the diffusion weighted and non-diffusion weighted images in this example. The data from the binary file "`S-all`" can then be read into the R session using the function `dtiData()`.

```
R> dwiobj <- dtiData(grad,"S-all",ddim)
```

All three arguments are required. `ddim` denotes the dimension vector of length 3 of the dataset, while the number N_{grad} of diffusion weighted images is implicitely given by the dimension of `grad`. `help(dtiData)` provides documentation of the function and more arguments, e.g. the choice of a ROI etc. The result `dwiobj` is an object of class "`dtiData`", see subsection 5.2 for details on the class definitions within the package.

```
R> dwiobj <- sdpar(dwiobj,interactive=TRUE)
```

is a function to interactively set a threshold that characterizes grey values in non-diffusion weighted images of voxel within the head and to estimates the parameters of the variance model (13). To assist the selection of a cut-off point densities of S_0 intensities are provided for the full data cube as well as three central subcubes of different size. The left mode of the densities is expected to correspond to voxel outside the head, i.e. a cut off point should be selected on the right of this first mode, see the vertical red line in Figure 2. If not called explicitely this function will be used by `dtiTensor()` in its non-interactive mode.

In the current version of the package diffusion weighted data is processed within the Dif-

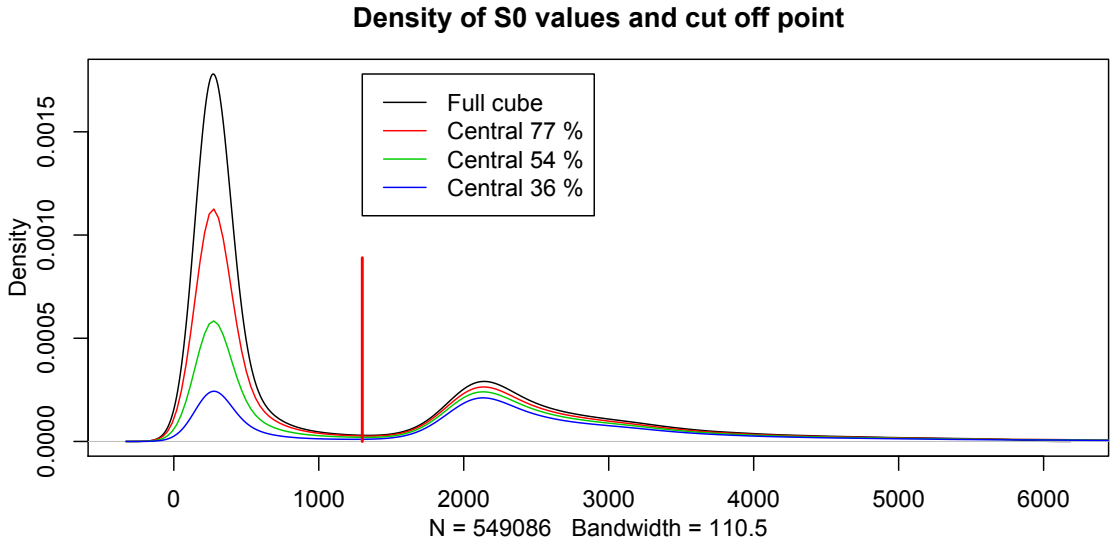

Figure 2: Selection of cut-off points for characterization of voxel within the head.

fusion Tensor Model. The method `dtiTensor` on any object of class "`dtiData`" estimates the diffusion tensor:

```
R> dtobj <- dtiTensor(dwiobj)
```

The standard method for the tensor estimation uses the non-linear model (10) described in section 2.3. However, the linearized model (9) can be used by specifying the argument `method=''linear''`. The resulting `dtobj` is an object of class `dtiTensor`, see subsection 5.2.

From the diffusion tensor a number of measures based on the eigenvalue representation of the tensor can be derived, like the trace, fractional anisotropy (FA), geodetic anisotropy (GA), shape parameters or the principal eigenvector defining the main diffusion direction, see section 2. The method `dtiIndices` on objects of class "`dtiTensor`" returns an object of class "`dtiIndices`" containing all these quantities:

```
R> dtind <- dtiIndices(dtobj)
```

Visualization of the results can be done using the generic `plot` functions on all three objects `dwiobj`, `dtobj`, and `dtind`, which show slices of the data, the tensor components, and color coded directional maps, respectively. If appropriate the `plot`-function returns an **adimpro**-image which can be further processed using functions from the **adimpro**-package [29]. 3D visualization for diffusion tensors and indices is provided by generic function `show3D` which makes use of the **rgl** package.

5.2 Objects and methods

The package implements S4-classes and -methods. The main class is ''dti'' from which no instances should be created. This class inherits to three subclasses ''dtiData'' for diffusion weighted data, ''dtiTensor'' for the estimated diffusion tensor objects and ''dtiIndices'' for tensor indices like fractional anisotropy. The class constructors provide consistency checks for these objects. Although not recommended, it is possible to access the slots directly. See `class ? dti` for documentation of the classes. For all subclasses methods for the generic functions `plot`, `print`, and `summary` have been implemented. Generic functions `extract` and ``[<-'' allow for extraction of information and index operation, respectively. See `methods ? <name>` for documentation of the methods and `help(<name>)` for information on the functions.

5.3 Data smoothing

Due to the high noise level in diffusion weighted data the package **dti** provides structural adaptive smoothing of diffusion weighted data as one of its main features. The method `dti.smooth()` implements the procedure described in the first sections of this paper. Since the algorithm directly smooths the diffusion weighted images it is only implemented for objects of class "`dtiData`".

`dti.smooth()` can be used with several parameter switches. One choice is between linear and non-linear tensor estimation as described in the section 2 of this paper. The default is set to non-linear estimation. Another choice is whether to include the correction for Rician bias, which defaults to "`TRUE`".

```
R> dtobjsmooth <- dti.smooth(dwiobj, hmax = 3, method = "linear",
                    rician = TRUE )
R> dtindsmooth <- dtiIndices(dtobjsmooth)
R> plot(dtindsmooth, slice = 30)
```

The main smoothing parameter is the maximum bandwidth `hmax` used for the iteration. It directly influences the amount of smoothness in the homogeneous regions of the data, and the complexity of the calculations via the number of iteration steps $k^\star$. Typical values are in the range of $2 - 4$ (voxel).

Other parameters are technical and include the degree of adaptation, the degree of regularization of the tensor estimates for small bandwidths, some display parameters, and choices about the variance model, in case of `method=''linear''`, to be used in the estimation

of the variance of the tensor estimates (see `help(dti.smooth)`). They are intended for expert use only.

5.4 Data visualization

For all subclasses of "`dti`" we implemented the generic `plot()`-function to visualize the objects in two dimensions. If appropriate the function returns an object of class `adimpro` for further processing with the functions of the package **adimpro**. For "`dtiData`" objects a slice of the selected diffusion weighted image is shown. For "`dtiTensor`" objects the tensor components of the selected slice are shown. No image is returned. For "`dtiIndices`" objects a color coded directional map is shown and returned as **adimpro** object. See the documention (`methods ? plot`) for details and left part of Figs. 4 and 5 for examples.

Based on the package **rgl** and OpenGL the package **dti** provides a 3D visualization with the method `show3d` for classes "`dtiTensor`" and "`dtiIndices`": For a tensor object ellipsoids are shown, while for the tensor indices lines are drawn with length corresponding to the FA-value, see Fig. 3.

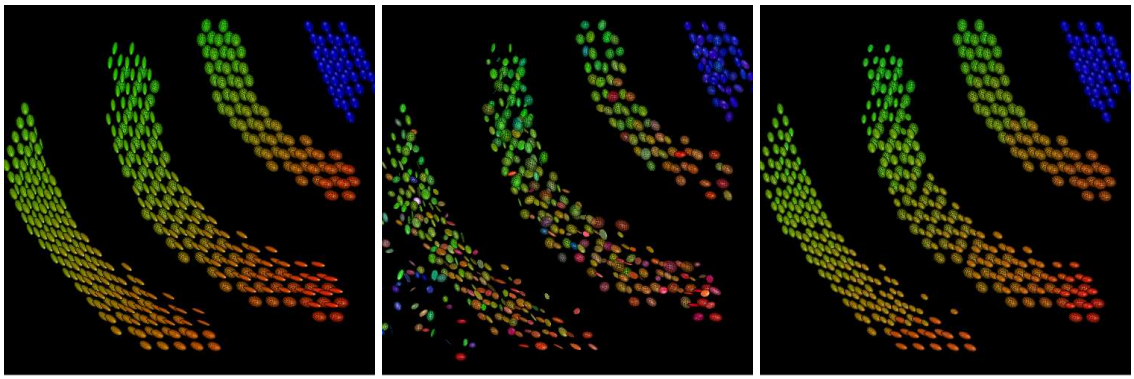

Figure 3: Artificial example from Ref.[15]: True tensor (left), voxelwise tensor estimates suffering from noise (center) and smoothed tensor estimates (right). A threshold of 0.3 for FA-values is used.

5.5 Experimental Data Examples

For illustration we use a DWI data set [32] made available by the NIH/NCRR Center for Integrative Biomedical Computing, P41-RR12553. These images are typical examples of diffusion tensor imaging of the brain. They contain twelve diffusion weighted volumes and one non-diffusion weighted ($b = 0$) reference volume. The data has a spatial resolution of 1.5 mm on each axis. The front of the head is at the top of the image. This scan goes from

the top of the head down to about the middle of the brain, below the corpus callosum, but above the eyes.

The DTI data was collected on a 3 Tesla MRI scanner in the W.M. Keck Laboratory for Functional Brain Imaging and Behavior by Dr. Andrew Alexander, Departments of Medical Physics and Psychiatry, University of Wisconsin, Madison, funding: NIH RO1 EB002012.

The Figures 4 and 5 illustrate some of the properties of the structural adaptive smoothing algorithm, namely the homogenization of the directional field within homogenous regions and the preservation of borders.

5.6 Interface to MedINRIA

In the current version of the package **dti** no fiber tracking algorithms are implemented. Therefore, we decided to provide an interface to MedINRIA [33] which runs on different major platforms. The function `tensor2medinria()` writes a NIfTI file with the tensor data in an appropriate format which can be read by MedINRIA.

```
R> tensor2medinria(dtobj, file= ...)
```

It is also possible to read a tensor object from MedINRIA given as NIfTI file for further processing like structure adaptive smoothing with the package **dti** in R:

```
R> dtobj <- medinria2tensor(file= ...)
```

5.7 Computation time and memory requirements

A word of caution concerning the memory usage of the implementation is in place here: DWI datasets are usually very large. A typical full brain dataset of matrix size 256×256 and about 70 slices measured at 30 diffusion gradient directions in 2-byte integer representation needs $256 \times 256 \times 70 \times 30 \times 2 \approx 262.5$ MBytes. The structure adaptive smoothing algorithm operates on *all* diffusion weighted images. In the current implementation these data are therefore kept in memory and it is usually not possible to run a full dataset on a Desktop computer. One has to restrict the analysis to a ROI if restricted memory is a problem.

We provide information on memory usage and CPU-time for various steps of the analysis and three DWI data sets in Table 1. Dataset 1 (first column) is small, with few gradient directions and without replicated non-diffusion weighted images. Datasets 2 (center) and 3 (third column) are characterized by a large number of gradients and replicated non-

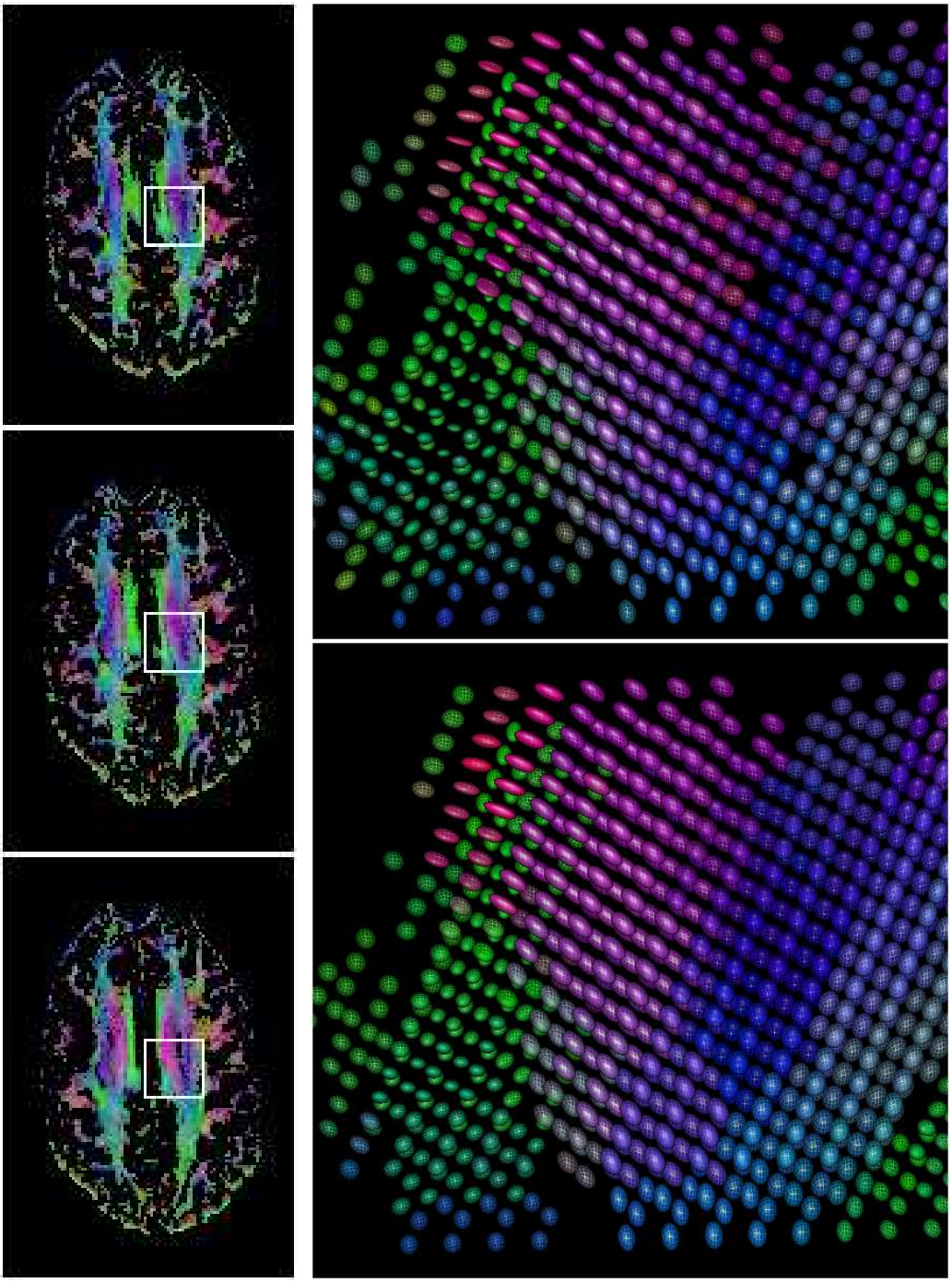

Figure 4: Real DWI data example: The left column shows the estimated color-coded directional map weighted with FA for slices 22-24 of the CIBC-dataset [32]. White squares mark the extend of the region used in the right column. There, the noisy (top) and smoothed (bandwidth 4, bottom) tensors are visualized. Structural adaptive smoothing apparently leads to a homogenization of the regions without blurring the structural borders.

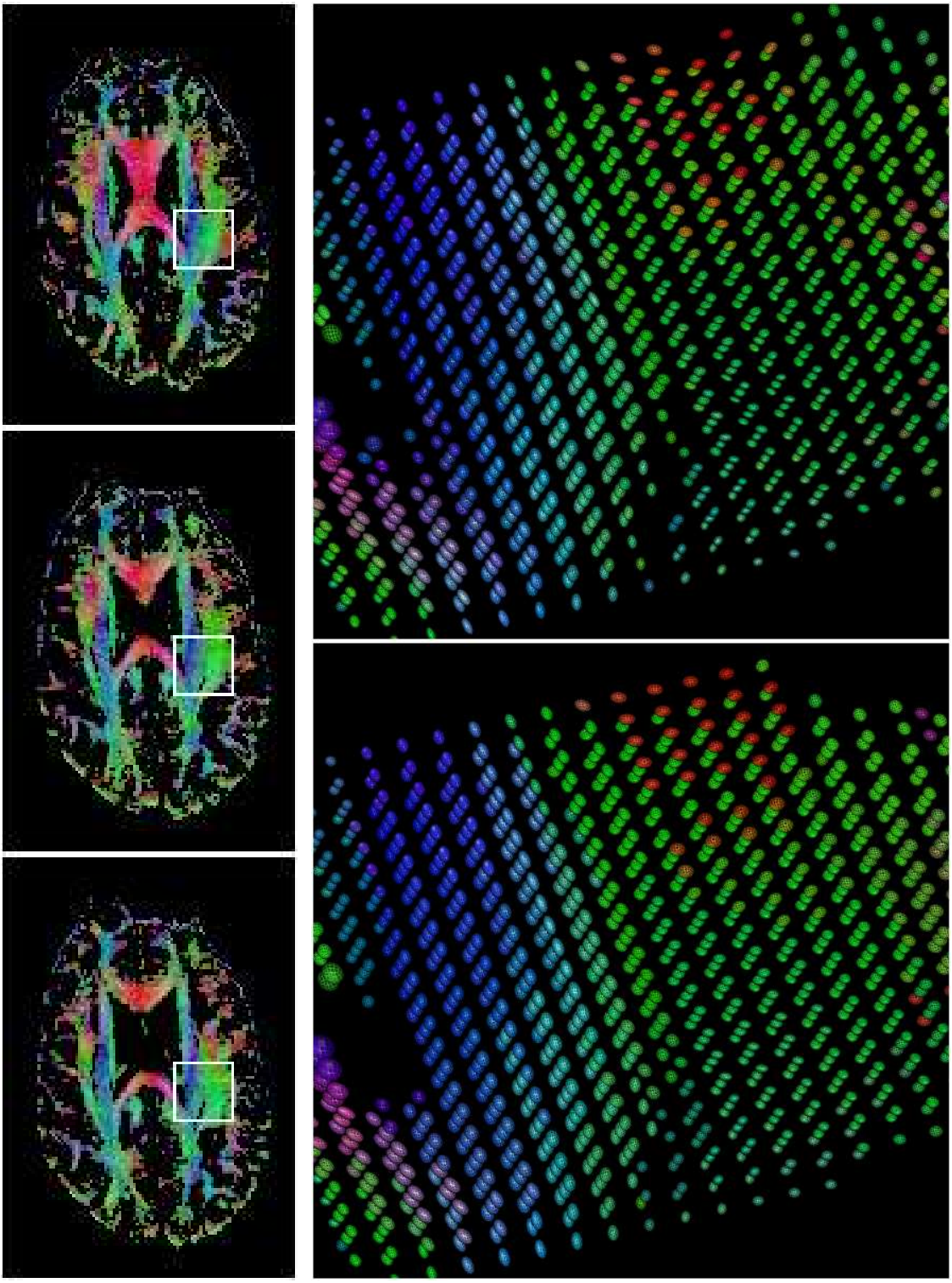

Figure 5: Results for a region within slices 28-30 of the CIBC-dataset [32]. See caption of Figure 4 for interpretation of content.

Table 1: Memory usage and computing time for three DWI data sets. The first column corresponds to the CIBC-dataset [32], the second to a data set kindly made available by A. Anwander, and a third data set kindly made available by H.U. Voss.

Dimensions	$101 \times 146 \times 38$	$72 \times 100 \times 50$	$146 \times 193 \times 47$
# gradient directions	13	201	150
Size of dtiData-object	27.8 MB	276.1 MB	757.8 MB
Size of dtiTensor-object	33.0 MB	20.8 MB	76.1 MB
Size of dtiTensor-object(smoothed)	41.6 MB	26.3 MB	96.3 MB
Size of dtiIndices-object	162 MB	24.7 MB	91 MB
Total memory (reported by gc())	525 MB	2144 MB	6048 MB
# voxel in mask	211235	255257	863134
CPU-time for sdpar	41 s	27 s	126 s
CPU-time for dtiTensor	16 s	118 s	156 s
CPU-time for dti.smooth ($h_{max} = 2$)	220 s	667 s	893 s
CPU-time for dti.smooth ($h_{max} = 4$)	415 s	1256 s	2140 s
CPU-time for dtiIndices	3.1 s	3.7 s	11 s
Mean N_i ($h_{max} = 2/4$)	3.67/16.5	1.20/1.87	1.55/4.32

diffusion weighted images. All datasets have been reduced in size to a cube containing all voxel within the head. Computations are restricted to voxel inside the head using a mask. The first two datasets have been processed on a PC equipped with an Intel(R) Core(TM)2 Duo E 6850 3 GHz and 4 GByte of memory. For the third dataset we used an Intel(R) Xeon(R) CPU 5160 3GHz and 24 GByte of memory. The operating system was OpenSuse 11.0 with R-version 2.7.1.

The values recorded for the mean sum of weights (mean of N_i) for the three data sets reflect the very different variability of the tensor estimates due to the varying number of gradients used.

5.8 Future plans

The diffusion tensor model does not appropriately describe diffusion weighted data, since it does not account for complex intravoxel structure (fiber crossings) and partial volume effects. We are therefore currently developing a structure adaptive algorithm for HARDI-data, which combines smoothing the diffusion weighted images, estimation of the ODFs or other angular distribution measures and fiber tracking in one algorithm.

We are open to suggestions for improving the package, as well as reports on bugs. We

hope the package will provide an easy-to-use basis for analysis of diffusion weighted data with R and will be useful for a broad audience.

Acknowledgments

This work is supported by the DFG Research Center MATHEON. It was made possible in part by software and a dataset from the NIH/NCRR Center for Integrative Biomedical Computing, P41-RR12553. The authors would like to thank A. Anwander at the Max Planck Institute for Human Cognitive and Brain Sciences (Leipzig, Germany) and H.U. Voss at the Citigroup Biomedical Imaging Center, Weill Cornell Medical College for providing diffusion-weighted MR datasets used within this paper. Furthermore, the authors would like to thank H.U. Voss for numerous intense and helpful discussions on Magnetic Resonance Imaging and related issues.

References

[1] D. LeBihan and E. Breton. Imagerie de diffusion in vivo par résonance magnétique nucléaire (invivo magnetic resonance imaging of diffusion). *C.R. Acad. Sci. Paris.*, 301:1109–1112, 1985.

[2] K. D. Merboldt, W. Hanicke, and J. Frahm. Self-diffusion NMR imaging using stimulated echoes. *J. Magn. Res.*, 64:479–486, 1985.

[3] D. G. Taylor and M. C. Bushell. The spatial mapping of translational diffusion coefficients by the NMR imaging technique. *Phys. Med. Biol.*, 30:345–349, 1985.

[4] H. Y. Carr and E. M. Purcell. Effects of diffusion on free precession in nuclear magnetic resonance experiments. *Physical Review*, 94:630–638, 1954.

[5] D. Le Bihan, J. F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat. Diffusion tensor imaging: Concepts and applications. *J. Magn. Res. Imag.*, 13:534–546, 2001.

[6] C. A. Clark, G. J. Barker, and P. S. Tofts. Magnetic resonance diffusion imaging of the human cervical spinal cord in vivo. *Magn. Reson. Med.*, 41:1269–1273, 1999.

[7] S. O. Rice. Mathematical analysis of random noise-conclusion. *Bell Systems Tech. J.*, 24:46–156, 1945.

[8] E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient. *J. Chem. Phys.*, 42:288–292, 1965.

[9] P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical J.*, 66:259–267, 1994.

[10] P. J. Basser, J. Mattiello, and D. LeBihan. Estimation of the effective self-diffusion *tensor* from the NMR spin echo. *J. Magn. Res.*, B(103):247–254, 1994.

[11] D. S. Tuch, R. M. Weisskoff, J. W. Belliveau, and V. J. Wedeen. High angular resolution diffusion imaging of the human brain. In *Proc. 7th Annual Meeting of ISMRM*, page p 321, 1999.

[12] L. R. Frank. Anisotropy in high angular resolution diffusion-weighted MRI. *Magn. Res. Med.*, 45:935–939, 2001.

[13] P. J. Basser and S. Pajevic. Statistical artefacts in diffusion tensor MRI (DT-MRI) caused by background noise. *Magn. Res. Med.*, 44(1):41–50, 2000.

[14] K. R. Hahn, S. Prigarin, S. Heim, and K. Hasan. Random noise in diffusion tensor imaging, its destructive impact and some corrections. In J. Weickert and H. Hagen, editors, *Visualization and Image Processing of Tensor Fields*, Mathematics and Visualization. Springer, Berlin, 2006.

[15] Karsten Tabelow, Jörg Polzehl, Vladimir Spokoiny, and Henning U. Voss. Diffusion tensor imaging: Structural adaptive smoothing. *NeuroImage*, 39:1763–1773, 2008.

[16] J. Polzehl and V. Spokoiny. Propagation-separation approach for local likelihood estimation. *Probab. Theory and Relat. Fields*, 135:335–362, 2006.

[17] J. Mattiello, P. J. Basser, and D. LeBihan. Analytical expressions for the b matrix in NMR diffusion imaging and spectroscopy. *J. Magn. Res. A*, 108:131–141, 1994.

[18] J. Mattiello, P. J. Basser, and D. LeBihan. The b matrix in diffusion tensor echo-planar imaging. *Magn. Res. Med.*, 37:292–300, 1997.

[19] P. Thomas Fletcher. *Statistical Variability in Nonlinear Spaces: Application to Shape Analysis and DT-MRI.* PhD thesis, University of North Carolina at Chapel Hill, 2004.

[20] I. Corouge, P.T. Fletcher, S. Joshi, S. Gouttard, and G. Gerig. Fiber tract-oriented statistics for quantitative diffusion tensor mri analysis. *Medical Image Analysis*, pages 786–798, 2006.

[21] C.-F. Westin, S. E. Maier, B. Khidhir, P. Everett, F. A. Jolesz, and R. Kikinis. Image processing for diffusion tensor magnetic resonance imaging. In *Medical Image Computing and Computer-Assisted Intervention*, Lecture Notes in Computer Science, pages 441–452, September 19–22 1999.

[22] A. L. Alexander, K. Hasan, G. Kindlmann, D. L. Parker, and J. S. Tsuruda. A geometric analysis of diffusion tensor measurements of the human brain. *Magn. Reson. Med.*, 44:283–291, 2000.

[23] D. K. Jones. When is a DT-MRI sampling scheme truly isotropic? *Proc. Intl. Soc. Magn. Reson. Med.*, 11:2118, 2003.

[24] D. K. Jones. The effect of gradient sampling schemes on measures derived from diffusion tensor MRI: A Monte Carlo study. *Magn. Res. Med.*, 51:807–815, 2004.

[25] P. B. Kinglsey. Introduction to diffusion tensor imaging mathematics: Part III. Tensor calculation, noise, simulations, and optimization. *Concepts Mag. Res. A*, 28A:155–179, 2006.

[26] Cheng Guan Koay, John D. Carew, Andrew L. Alexander, Peter J. Basser, and M. Elizabeth Meyerand. Investigation of anomalous estimates of tensor-derived quantities in diffusion tensor imaging. *Magnetic Resonance in Medicine*, 55:930936, 2006.

[27] Cheng Guan Koay, Lin-Ching Chang, Carlo Pierpaoli, and Peter J. Basser. Error propagation framework for diffusion tensor imaging via diffusion tensor representations. *IEEE Transactions on Medical Imaging*, 26(8):1017–1034, August 2007.

[28] H. Schwetlick. *Numerische Lösung nichtlinearer Gleichungen.* Deutscher Verlag der Wissenschaften, Berlin, 1979.

[29] Jörg Polzehl and Karsten Tabelow. Adaptive smoothing of digital images: The R package adimpro. *J. Statist. Software*, 19:1–17, 2007.

[30] Saurav Basu, Thomas Fletcher, and Ross Whitaker. Rician noise removal in diffusion tensor MRI. In *Medical image computing and computer-assisted intervention*, LNCS 4190, pages 117–125. MICCAI, 2006.

[31] K. Tabelow, J. Polzehl, H. U. Voss, and V. Spokoiny. Analyzing fMRI experiments with structural adaptive smoothing procedures. *Neuroimage*, 33:55–62, 2006.

[32] CIBC, 2008. Data Sets: NCRR Center for Integrative Biomedical Computing (CIBC) data set archive. URL=http://www.sci.utah.edu/cibc/software/login_datasets.html.

[33] Pierre Fillard Jean-Christophe Souplet and Nicolas Toussaint. *Medical Image Navigation and Research Tool by INRIA (MedINRIA) Tutorial v2.0.* INRIA Sophia Antipolis - Research Project ASCLEPIOS., 2004, route des Lucioles - BP 93. 06902 Sophia Antipolis Cedex France, June 2007.